

**Instituto Tecnológico de Costa Rica**

**Escuela de Ingeniería en Electrónica**



**Entorno de Simulación para el Análisis de Fallas en  
Instalaciones Fotovoltaicas con Python**

Informe de Proyecto de Graduación para optar por el título de  
Ingeniería en Electrónica con el grado académico de  
Licenciatura

Tania Pamela Martínez Abarca

Carné: 2016138274

Profesor asesor:

Prof. Ing. Hugo Sánchez Ortiz

CARTAGO, I SEMESTRE 2021

**INSTITUTO TECNOLÓGICO DE COSTA RICA**

**ESCUELA DE INGENIERÍA ELECTRÓNICA**

**TRABAJO FINAL DE GRADUACIÓN**

**ACTA DE APROBACIÓN**

**Defensa del Trabajo Final de Graduación  
Requisito para optar por el título de Ingeniero en Electrónica  
Grado Académico de Licenciatura  
Instituto Tecnológico de Costa Rica**

El Tribunal Evaluador aprueba la defensa del Trabajo Final de Graduación denominado Entorno de Simulación para el Análisis de Fallas en Instalaciones Fotovoltaicas con Python, realizado por la señorita Tania Pamela Martínez Abarca y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador

---

Ing. Sergio Morales Hernández

Profesor lector

---

Ing. Carlos Mauricio Segura Quirós

Profesor lector

---

Ing. Hugo Sánchez Ortiz

Profesor asesor

Cartago, 28 de Junio, 2021

### **Declaración de Autenticidad**

Declaro que el presente Trabajo de Graduación ha sido realizado, en su totalidad, por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas. En consecuencia, asumo responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe.



---

Tania Pamela Martínez Abarca

Cédula: 117170304

Cartago, Junio 2021

This work is licensed under a Creative Commons  
“Attribution-NonCommercial-ShareAlike 3.0 Unported” li-  
cense.



## Resumen

La energía solar fotovoltaica ha sido una de las energías renovables pioneras durante décadas. La evolución que ha tenido la industria solar fotovoltaica hasta ahora ha sido notable gracias al acelerado avance de estas tecnologías renovables, con varios hitos logrados en los últimos años en términos de instalaciones y reducciones de costos, así como el establecimiento de asociaciones encargadas de promover el uso y desarrollo de la energía solar. Este tipo de energía renovable proyecta que seguirá siendo una opción primordial en las próximas décadas.

Con el pasar del tiempo se evidencia que las instalaciones de sistemas fotovoltaicos se encuentran cada vez más presentes en distintos tipos de sectores y en distintas dimensiones. Al hacer uso de estas tecnologías cada vez más se espera que estas sean altamente confiables y dignas de inversión ya que son las encargadas de la generación de energía.

Las instalaciones fotovoltaicas cuentan con un número considerable de paneles solares que están expuestos a condiciones ambientales que generan fallas en estos, aparte de las que se pueden generar por otros factores. Por esta razón, en este proyecto se desarrolla un entorno de simulación capaz de simular los efectos de fallas eléctricas, de ensuciamiento y sombreado que afectan el funcionamiento de las instalaciones fotovoltaicas; con la finalidad de inferir la respuesta que tendrán estas tecnologías a futuro en un escenario práctico.

**Palabras clave:** Energía solar, Instalaciones Fotovoltaicas, Panel Solar, Inversor, Bibliotecas, Simulador.

## Abstract

Photovoltaic solar energy has been one of the pioneering renewable energies for decades. The evolution that the solar photovoltaic industry has had so far has been remarkable thanks to the accelerated advance of these renewable technologies, with several milestones achieved in recent years in terms of installations and cost reductions, as well as the establishment of associations in charge of promoting the use and development of solar energy. This type of renewable energy projects that it will continue to be a prime option for decades to come.

With the passing of time it is evident that the installations of photovoltaic systems are increasingly present in different types of sectors and in different dimensions. By making use of these technologies, it is increasingly expected that they are highly reliable and worthy of investment since they are in charge of generating energy.

Photovoltaic installations have a considerable number of solar panels that are exposed to environmental conditions that generate failures in them, apart from those that can be generated by other factors. For this reason, this project develops a simulation environment capable of simulating the effects of electrical faults, soiling and shading that affect the operation of photovoltaic installations; in order to infer the response that these technologies will have in the future in a practical scenario.

**Keywords:** Solar energy, Photovoltaic Installations, Solar Panel, Inverter, Libraries, Simulator.

*A mis amados padres, Erika y Yehudi*

## Agradecimientos

Primero quiero agradecer a Dios, por darme la oportunidad de haber llegado hasta donde estoy y siempre mostrarme el camino cuando más perdida me sentía.

A mis papás, Erika y Yehudi, porque sin ellos nada de esto hubiese sido posible, mi mayor admiración hacia ustedes, siempre serán mi ejemplo a seguir. Mi motivación en el día a día y las ganas de seguir adelante siempre será gracias a los dos. Gracias por siempre estar presentes y apoyarme en los buenos y malos momentos, por tenderme en muchas ocasiones hasta el hombro cuando sentía que no daba más. Muchas gracias por las miles de comidas preparadas, por recogerme a altas horas de la noche aún estando agotados por el trabajo, nunca se los terminaré de agradecer. Para ustedes todo este esfuerzo, que aunque no lo parezca son los coautores de este proyecto.

Quiero agradecerles a mis hermanas Maryel y Denisse, por escucharme en tantos momentos de estrés y por demostrarme en cada ocasión lo capaz que soy. Gracias por enseñarme que la hermana mayor no siempre es la que enseña. A mis demás familiares también les quiero agradecer el apoyo incondicional y los consejos que me dieron a lo largo de mi vida, todos aportaron cosas positivas a este momento.

Gracias a mis amigos y compañeros de la universidad, porque cada uno de ellos forma parte importante de esta etapa, realmente la hicieron inolvidable. En especial a Luis, por siempre estar a mi lado recordándome lo valiosa que soy y ayudándome a ser no solo una mejor profesional, sino que también una mejor persona.

Finalmente, quiero agradecer a la escuela de Ingeniería Electrónica por ser mi segunda casa durante estos cinco años, y a todos los profesores que me topé en el camino. Quiero agradecerle en particular a mis profesores Carlos Meza y Hugo Sánchez, por todos los consejos dados y por el conocimiento transmitido para culminar de la mejor manera este proyecto.

Tania Pamela Martínez Abarca  
Cartago, 2021





---

# ÍNDICE GENERAL

<b>ÍNDICE DE FIGURAS</b>	<b>10</b>
<b>ÍNDICE DE TABLAS</b>	<b>12</b>
<b>1 Introducción</b>	<b>13</b>
1.1. Entorno . . . . .	13
1.2. Planteamiento del problema . . . . .	15
1.3. Síntesis del problema . . . . .	17
1.4. Enfoque de la solución . . . . .	17
1.5. Objetivos . . . . .	18
1.5.1. Meta . . . . .	18
1.5.2. Objetivo General . . . . .	18
1.5.3. Objetivos Específicos . . . . .	18
1.6. Estructura del documento . . . . .	18
<b>2 Fundamentos Teóricos</b>	<b>20</b>
2.1. Sistemas Fotovoltaicos . . . . .	20
2.2. Sistemas Causales y Acausales . . . . .	26
2.3. Lenguaje de descripción de hardware . . . . .	27
2.4. Comparación de herramientas . . . . .	28
2.5. Python . . . . .	29
<b>3 Arquitectura del sistema de simulación</b>	<b>31</b>
3.1. Configuraciones a simular . . . . .	31
3.2. Entorno de simulación . . . . .	33
3.2.1. Características del sistema . . . . .	33
3.2.2. Comparación de bibliotecas . . . . .	34

3.2.3.	Bibliotecas utilizadas . . . . .	35
3.2.4.	Desarrollo del entorno de simulación . . . . .	37
3.2.4.1.	Escenarios simulados . . . . .	40
3.2.4.2.	Funcionamiento del programa . . . . .	44
<b>4</b>	<b>Pruebas y verificación del entorno de simulación</b>	<b>46</b>
4.1.	Ejecución del programa . . . . .	46
4.2.	Cálculo de tiempo promedio de simulación y porcentaje de diferencia entre tensiones . . . . .	49
4.3.	Simulación de la configuración del Inversor Central . . . . .	50
4.4.	Simulación de configuración del Inversor de Cadena (String) . . . . .	55
4.5.	Síntesis . . . . .	60
<b>5</b>	<b>Conclusiones y recomendaciones</b>	<b>64</b>
5.1.	Conclusiones . . . . .	64
5.2.	Recomendaciones . . . . .	64
	<b>Bibliografía</b>	<b>66</b>
<b>A</b>	<b>Abreviaturas</b>	<b>70</b>

---

# ÍNDICE DE FIGURAS

1.1. Instalación de paneles fotovoltaicos en Instituto Tecnológico de Costa Rica.	14
1.2. Demanda anual y proyecciones de energía solar fotovoltaica por regiones, 2018-2050. . . . .	15
2.1. Composición de panel solar. . . . .	21
2.2. Curva característica I-V de un panel fotovoltaico . . . . .	21
2.3. Curvas características I-V dependientes de la radiación solar. . . . .	22
2.4. Curvas características I-V dependientes de la temperatura. . . . .	23
2.5. Esquema de Sistema Conectado a Red . . . . .	23
2.6. Esquema de Inversor Central . . . . .	24
2.7. Esquema de Inversor de Cadena . . . . .	24
3.1. Inversor Central . . . . .	32
3.2. Inversor de Cadena . . . . .	32
3.3. Secuencia de pasos a desarrollar . . . . .	37
3.4. Esquema de la composición del archivo utilitario. . . . .	38
3.5. Esquema de la composición del archivo generador de gráficas. . . . .	38
3.6. Esquema de la composición del archivo creador de circuito. . . . .	39
3.7. Diagrama de Componentes . . . . .	39
3.8. Diagrama de Casos de Uso . . . . .	40
3.9. Configuración Ideal de Inversor Central . . . . .	41
3.10. Configuración Ideal de Inversor de Cadena . . . . .	41
3.11. Representación de panel fotovoltaico . . . . .	42
3.12. Configuración Real de Inversor Central . . . . .	42
3.13. Configuración Real de Inversor de Cadena . . . . .	42
3.14. Falla de cortocircuito en panel para las configuraciones de Inversor Central e Inversor de Cadena. . . . .	43

3.15. Falla de circuito abierto en panel para las configuraciones de Inversor Central e Inversor de Cadena. . . . .	43
3.16. Diagrama de flujo de uso del sistema . . . . .	45
4.1. Ejemplo de ejecución sin fallos del Entorno de Simulación. . . . .	46
4.2. Ejemplo de gráfica generada en una ejecución sin fallos del Entorno de Simulación. . . . .	47
4.3. Solicitud de selección de fuentes para el control de interruptores. . . . .	48
4.4. Ejemplo de correcto almacenamiento de variables. . . . .	48
4.5. Error en el entorno de simulación . . . . .	49
4.6. Tensión de salida Escenario 1 - Inversor Central . . . . .	51
4.7. Tensión de salida Escenario 2 - Inversor Central . . . . .	52
4.8. Tensión de salida Cortocircuito - Inversor Central . . . . .	53
4.9. Tensión de salida Circuito Abierto - Inversor Central . . . . .	54
4.10. Tensión de salida Sombreado/Ensuciamiento - Inversor Central . . . . .	54
4.11. Tensión de salida Escenario 1 - Inversor de Cadena . . . . .	56
4.12. Tensión de salida Escenario 2 - Inversor de Cadena . . . . .	57
4.13. Tensión de salida Cortocircuito - Inversor de Cadena . . . . .	58
4.14. Tensión de salida Circuito Abierto - Inversor de Cadena . . . . .	58
4.15. Tensión de salida Sombreado/Ensuciamiento - Inversor de Cadena . . . . .	59
4.16. Inversor Central - Tensiones de salida en cada escenario de estudio. . . . .	60
4.17. Inversor de Cadena - Tensiones de salida en cada escenario de estudio. . . . .	61
4.18. Inversor Central - Tiempos de simulación en cada escenario de estudio. . . . .	62
4.19. Inversor de Cadena - Tiempos de simulación en cada escenario de estudio. . . . .	62

---

## ÍNDICE DE TABLAS

2.1. Tipos de fallas en sistemas fotovoltaicos. . . . .	25
2.2. Comparación cualitativa entre distintas herramientas. . . . .	28
3.1. Características del sistema. . . . .	33
3.2. Posibles bibliotecas para la simulación de sistemas fotovoltaicos. . . . .	34
4.1. Tabla resumen Inversor Central - Escenario 1 . . . . .	51
4.2. Tabla resumen Inversor Central - Escenario 2 . . . . .	52
4.3. Tabla resumen Inversor Central - Escenario 3 Cortocircuito . . . . .	53
4.4. Tabla resumen Inversor Central - Escenario 3 Circuito abierto . . . . .	54
4.5. Tabla resumen Inversor Central - Escenario 3 Sombreado/Ensuciamiento .	55
4.6. Tabla resumen Inversor de Cadena - Escenario 1 . . . . .	56
4.7. Tabla resumen Inversor de Cadena - Escenario 2 . . . . .	57
4.8. Tabla resumen Inversor de Cadena - Escenario 3 Cortocircuito . . . . .	58
4.9. Tabla resumen Inversor de Cadena - Escenario 3 Circuito abierto . . . . .	59
4.10. Tabla resumen Inversor de Cadena - Escenario 3 Sombreado/Ensuciamiento	60

# Introducción

## 1.1. Entorno

Las instalaciones de paneles fotovoltaicos han experimentado avances significativos en los últimos años, esto gracias a que la energía eléctrica generada puede ser utilizada en infinidad de aplicaciones, ya sea a gran o pequeña escala. Además, es importante recalcar que la energía que captan dichas instalaciones es renovable, por ende no contribuye a la generación de gases de efecto invernadero. Es fundamental dar seguimiento a los parámetros que brindan los sistemas fotovoltaicos porque contienen información relevante que puede ser utilizada para poder tener un control adecuado, y así prevenir algún tipo de falla en el funcionamiento. Hay que tener presente que la fuente de energía es inagotable y no contamina, colaborando así con el desarrollo sostenible. [1]

Es necesario aclarar que los paneles fotovoltaicos producen energía continua, y por medio de dispositivos llamados inversores se convierte esta corriente continua producida en corriente alterna. El tipo de inversor que se necesita depende de los objetivos y requisitos del proyecto. Además, este debe de cumplir todos los lineamientos del proveedor de electricidad actual por seguridad y protección, ya que los inversores en red se requieren por ley apagar durante un corte de suministro eléctrico. [2]

El Instituto Tecnológico de Costa Rica posee un laboratorio llamado SESLab (Laboratorio de Sistemas Eléctricos para la Sostenibilidad), el cual es un laboratorio de investigación aplicada con un enfoque en lo que es la evaluación, análisis y desarrollo de sistemas electrónicos que son utilizados en aplicaciones carbono neutral y de bajo impacto ambiental como sistemas de energía renovable, transporte cero carbono y medio ambiente. [3]

Uno de los proyectos de SESLab que ha tenido relevancia, junto con otras entidades, es el complejo solar ubicado en el Instituto Tecnológico de Costa Rica, específicamente en el campus de Cartago. Este complejo se compone de 1200 paneles (abarcando una

extensión de más de 2400 metros cuadrados), baterías en las cuales se almacena una parte de la energía generada, en donde esta generación de energía es de aproximadamente 400 kW. La planta solar produce del 25 al 30 % del consumo de la energía eléctrica anual en el campus de Cartago, ocasionando a su vez un ahorro económico considerable en el consumo eléctrico del TEC. Esta instalación conduce al TEC a formar parte de las instituciones líderes en investigación y desarrollo de este tipo de tecnologías energéticas, y al mismo tiempo motivando tanto a estudiantes, funcionarios y demás empresas a adquirir este tipo de tecnologías con el propósito de disminuir el impacto ambiental. En la figura 1.1 se facilita la visualización del complejo solar en el TEC. [4]



Figura 1.1: Instalación de paneles fotovoltaicos en Instituto Tecnológico de Costa Rica. [4]

A nivel tanto nacional como internacional, las instalaciones de sistemas fotovoltaicos ya no son restringidas a una escala pequeña, sino que estas han tomado fuerza para poder ser aplicadas en sistemas de mayor extensión como en el abastecimiento de ciudades rurales o en sistemas de señalización. [5]

En todas las regiones del mundo se ha experimentado un aumento en la utilización de instalaciones fotovoltaicas tanto a nivel industrial como en aplicaciones pequeñas, esto en gran medida gracias a las políticas de apoyo, principalmente incentivos fiscales para un mayor impacto de la actividad en los países, la caída de los costes y ahorros en las tarifas de electricidad pagadas por los usuarios. [6]

En la figura 1.2 se visualiza las perspectivas en el aumento de gigavatios de las instalaciones anuales a nivel global según valores históricos y proyecciones futuras basadas en las estadísticas de energía renovable de IRENA:



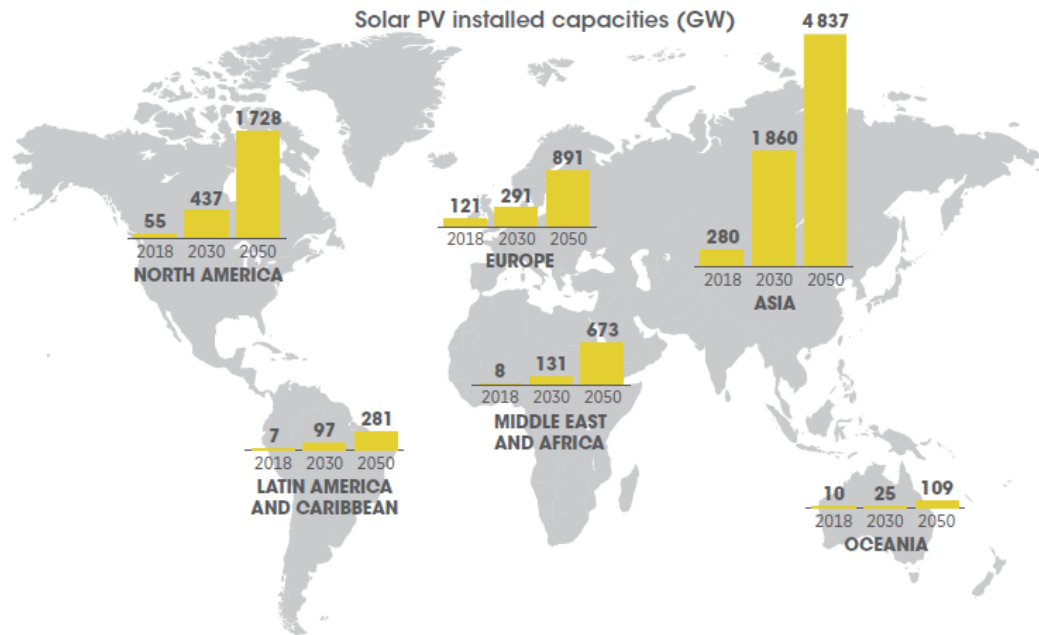


Figura 1.2: Demanda anual y proyecciones de energía solar fotovoltaica por regiones, 2018-2050.

[6]

Como se observó en la figura anterior, el crecimiento que han tomado las instalaciones fotovoltaicas a nivel global es relevante. Por lo tanto, es importante dar seguimiento a las fallas que se pueden generar en estos sistemas para asegurar el buen funcionamiento y al mismo tiempo mejorar la confiabilidad de estos. Asimismo, más usuarios alrededor del mundo estarían interesados en invertir en las instalaciones fotovoltaicas.

## 1.2. Planteamiento del problema

Las instalaciones de sistemas fotovoltaicos, con el pasar de los años buscan abastecer mayor cantidad de comunidades, aplicaciones con un alto consumo de energía o incluso generar electricidad para venderla a la red eléctrica. Cada una de las instalaciones posee muchos paneles conectados entre sí, en el presente existen escasas herramientas que pueden colaborar al análisis de fallas a nivel físico en los paneles haciendo esta tarea menos tediosa.

Para tener una visión más clara de la problemática presentada, se expone un ejemplo de la inmensidad que estas instalaciones pueden tener. El complejo en construcción de energía solar ubicado en Dubai llamado “Mohammed Bin Rashid Al Maktoum” pretende abarcar aproximadamente 200 kilómetros cuadrados. Se aspira a generar más de 5000 MW de electricidad, con lo que se podría abastecer a 1,3 millones de hogares y esto conlleva una responsabilidad muy grande, ya que se quiere evitar que existan

interrupciones en la generación de energía para el abastecimiento correcto. [7]

Ahora bien, como se mencionó anteriormente, hoy en día existe un uso limitado de herramientas para el estudio del análisis de fallas en instalaciones fotovoltaicas, etapa que es completamente necesaria para poder agilizar el proceso cuando este llegue a hacerse de forma práctica en un futuro y para validar lo propuesto en la teoría.

Hay que tomar en cuenta que existen implementaciones de sistemas fotovoltaicos en lugares alejados y de difícil acceso. En Costa Rica, se ha logrado dotar de energía solar a 296 familias de territorios indígenas de la zona sur, por lo que el uso de herramientas de simulación que permitan observar los efectos de posibles fallas en los paneles favorece el mantenimiento y resultados positivos para un normal funcionamiento del sistema. [8]

Lo que se desea es tener un correcto y continuo funcionamiento de los sistemas fotovoltaicos, por lo tanto es necesario que por medio de simulaciones se pueda conocer el comportamiento de los paneles cuando están expuestos a estos escenarios. Dentro de las fallas que pueden tener los paneles son:

- Eléctricas: Las que se tomarán en cuenta son el cortocircuito y el circuito abierto. Cuando se expone de forma continua el módulo a los rayos ultravioleta de la radiación solar, la resina que protege las células fotovoltaicas se deteriora, liberando de agentes que destruyen los contactos de metal oxidante. Estas causan la falta de producción de energía eléctrica. [9]
- Ensuciamiento y Sombreado: Estas fallas son producidas por la limitación en espacio u orientación y el asentamiento de partículas, algo fuera del control humano. Dentro de sus consecuencias son la afectación el rendimiento del módulo fotovoltaico. [9]

En un contexto real o físico estos problemas pueden suspender la producción de energía eléctrica de manera completa u ocasionar un hueco de tensión. Estas fallas se dan ya que los paneles al estar expuestos constantemente a la radiación solar dañan la resina de protección celular, provocando que sustancias oxidantes liberadas deterioren los contactos metálicos de los paneles. Cuando esto ocurre es ideal sustituir el panel fotovoltaico casi inmediatamente, ya que al estar conectada en serie con otras celdas afecta negativamente la producción energética de todo el sistema y su eficiencia. [10]

Las fallas en los sistemas fotovoltaicos también reflejan pérdidas económicas por el tiempo de inactividad del sistema, la sustitución/reparación de componentes y evaluación de la efectividad de las medidas de mitigación. De acuerdo al informe del proyecto, *Identificación de riesgos técnicos en la cadena de valor fotovoltaica y cuantificación del impacto económico* [11], utilizaron una metodología para evaluar el impacto económico de las fallas, resultando en que dentro de los diez principales riesgos técnicos para los módulos fotovoltaicos son el ensuciamiento, sombreado y fallas

dentro de la instalación. Al final del día, si existe una baja confiabilidad en el sistema fotovoltaico esto se traduce en menor atracción para la inversión. [11]

El hacer uso de una herramienta de simulación capaz de inferir el comportamiento o el efecto de los paneles exponiéndolos a fallas, ayuda a procurar que cuando se quieran hacer conexiones en un escenario real con otros dispositivos no se desencadenen daños a éstos por un efecto dominó. De modo que, si una sección de la instalación fotovoltaica presenta fallas las consecuencias no se den a una mayor escala. Esto no solo puede ser funcional para instalaciones de menor escala como la que se encuentra en el TEC de Cartago, sino que también para aquellas que comprenden un área mucho mayor.

No obstante, a pesar de existir herramientas como lo es *LTSpice*, *NI Multisim*, entre otras; que son utilizadas en distintos laboratorios como en el SESLab para la utilización de modelos gráficos, limitan el análisis de los fallos. Por este motivo, se crea la necesidad de desarrollar una herramienta más dinámica y más manipulable que permita hacer uso de recursos externos y análisis con un nivel matemático superior que satisfaga las necesidades de la investigación, y con ello, contribuir de manera adecuada a la industria fotovoltaica y al desarrollo de economías verdes.

### **1.3. Síntesis del problema**

Falta de dinamismo, capacidad limitada de análisis complejos y tiempos de ejecución prologandos en las herramientas de simulación disponibles con la finalidad de simular los efectos generados al aplicar fallas en circuitos equivalentes de sistemas fotovoltaicos.

### **1.4. Enfoque de la solución**

Las instalaciones de gran magnitud proporcionan una cantidad de energía importante a una zona específica, estas instalaciones están compuestas por muchos paneles que cubren una porción de terreno extensa. Lo que se desea es observar el comportamiento de distintas configuraciones de instalaciones fotovoltaicas conectadas a red a nivel de simulación cuando estas son sometidas a ciertas condiciones de error. Para esto se necesita un entorno de simulación capaz de emular estas configuraciones de paneles en donde se puede generar de forma intencional fallas, al igual que pueda interactuar con otros archivos capaces de brindar mayor información y poder introducir ecuaciones o modelos complejos a la herramienta con la finalidad de generar reportes más robustos. Obteniendo variables de salida y gráficas de la configuración cuando está expuesta a una falla eléctrica, de ensuciamiento o sombreado se puede tomar como base para inferir la

respuesta que tendrá a futuro si se llega a presentar algunos de los problemas anteriores en un contexto práctico.

## 1.5. Objetivos

### 1.5.1. Meta

Generar una aplicación de simulación que permita examinar casos de error previamente modelados con la finalidad de servir como base en la identificación de módulos dañados en el sistema fotovoltaico.

### 1.5.2. Objetivo General

- Desarrollar un entorno de simulación capaz de visualizar el comportamiento de paneles fotovoltaicos al ser expuestos a fallas eléctricas, de ensuciamiento y sombreado.

### 1.5.3. Objetivos Específicos

- Diseñar un entorno de simulación que permita representar distintas configuraciones de instalaciones fotovoltaicas conectadas a red.

Indicador: Capacidad de simular configuraciones por medio de bibliotecas de *Python*.

- Programar distintos escenarios de fallas para analizar su efecto en las configuraciones propuestas.

Indicador: Capacidad de emular situaciones de falla en el entorno de simulación propuesto.

- Validar el funcionamiento de la simulación desarrollada comparando con resultados de la herramienta *LTSpice*.

Indicador: Porcentaje de diferencia menor a 5 % en las variables eléctricas.

## 1.6. Estructura del documento

La estructura del siguiente documento está distribuida de la siguiente manera:

- Capítulo 1: Este es el capítulo introductorio, en el cuál se da una visión general del porqué se quiere desarrollar el proyecto; así como los objetivos del mismo.

- Capítulo 2: En este capítulo se presentan algunas definiciones necesarias para un mayor entendimiento de los distintos componentes del proyecto.
- Capítulo 3: Se muestra todo lo relacionado a la arquitectura del sistema de simulación, tanto de las configuraciones a simular como el funcionamiento del sistema simulado.
- Capítulo 4: Dentro de este capítulo se añaden las pruebas del funcionamiento del entorno de simulación desarrollado donde muestra los resultados obtenidos.
- Capítulo 5: Finalmente, este capítulo contiene las conclusiones y recomendaciones del trabajo realizado.

---

## Fundamentos Teóricos

En este capítulo se encuentran definiciones necesarias para un mayor entendimiento del desarrollo del proyecto. Cada una de las secciones muestra elementos importantes a tomar en cuenta para entender qué son los sistemas fotovoltaicos, cuáles fallas se pueden encontrar dentro de este, qué determina la causalidad en un sistema y la elección de la herramienta de uso que provee mayor aprovechamiento para el proyecto.

### 2.1. Sistemas Fotovoltaicos

Los sistemas fotovoltaicos son los cuáles tienen la capacidad de transformar o realizar la conversión directa de la radiación solar en electricidad. El uso de la energía solar fotovoltaica puede llegar tanto a satisfacer la escasez energética de quienes no disponen de la red eléctrica, hasta generar energía a la red eléctrica. [12] Estos sistemas realizan la conversión por medio de paneles fotovoltaicos (conjunto de células solares o fotovoltaicas) los cuales se conforman de materiales semiconductores, mayoritariamente silicio, y recubiertos por vidrio que permite la producción del efecto fotovoltaico al dejar pasar la radiación solar que a su vez minimiza las pérdidas. Los elementos que componen un panel fotovoltaico son:

- Marco de aluminio
- Cristal templado
- Encapsulado
- Células fotovoltaicas
- Cubierta de protección
- Caja de conexiones

En la figura 2.1 puede apreciarse la composición de un panel fotovoltaico:

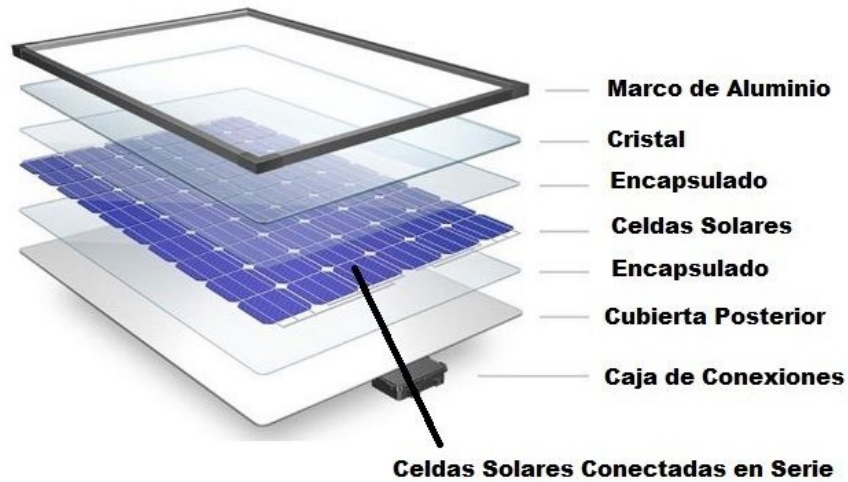


Figura 2.1: Composición de panel solar.  
[13]

Las células fotovoltaicas que componen el panel pueden estar conectadas tanto en serie como en paralelo, de acuerdo a la tensión (colocar células en serie hasta alcanzar el valor necesario) y corriente (colocación de células en paralelo) requerida por el panel. Este comportamiento eléctrico es representado por medio de la curva característica I-V del panel solar, la cual se puede visualizar en la figura 2.2. Hay que tomar en cuenta que las condiciones ambientales consideradas son una radiación solar de  $1000 \text{ W/m}^2$  y una temperatura ambiente de  $25^\circ\text{C}$ . [13]

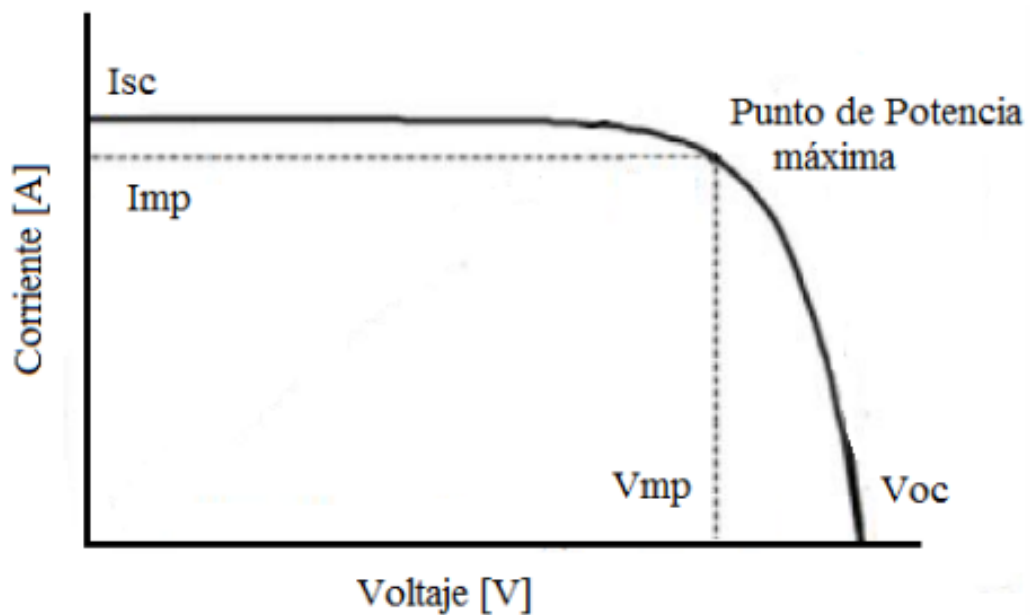


Figura 2.2: Curva característica I-V de un panel fotovoltaico  
[13]

Los parámetros que componen la curva característica son:

- Intensidad de cortocircuito ( $I_{sc}$ ): es la máxima corriente medida en el panel cuando la tensión de salida es cero.
- Tensión de circuito abierto ( $V_{oc}$ ): Es la máxima tensión medida en el panel cuando la corriente es cero.
- Intensidad en el punto de máxima potencia ( $I_{mp}$ ): Corresponde al valor de la corriente cuando la potencia producida por el panel es la máxima.
- Tensión en el punto de máxima potencia ( $V_{mp}$ ): Es el valor de la tensión cuando la potencia producida por el módulo es la máxima.
- Potencia máxima ( $P_{max}$ ): Máxima potencia que producirá el panel cuando la corriente y tensión tienen sus valores máximos.

Existe una variación en la curva característica cuando las condiciones ambientales no están normalizadas, como cuando se presentan variaciones en la temperatura (figura 2.3) y la irradiancia (figura 2.4). [14]

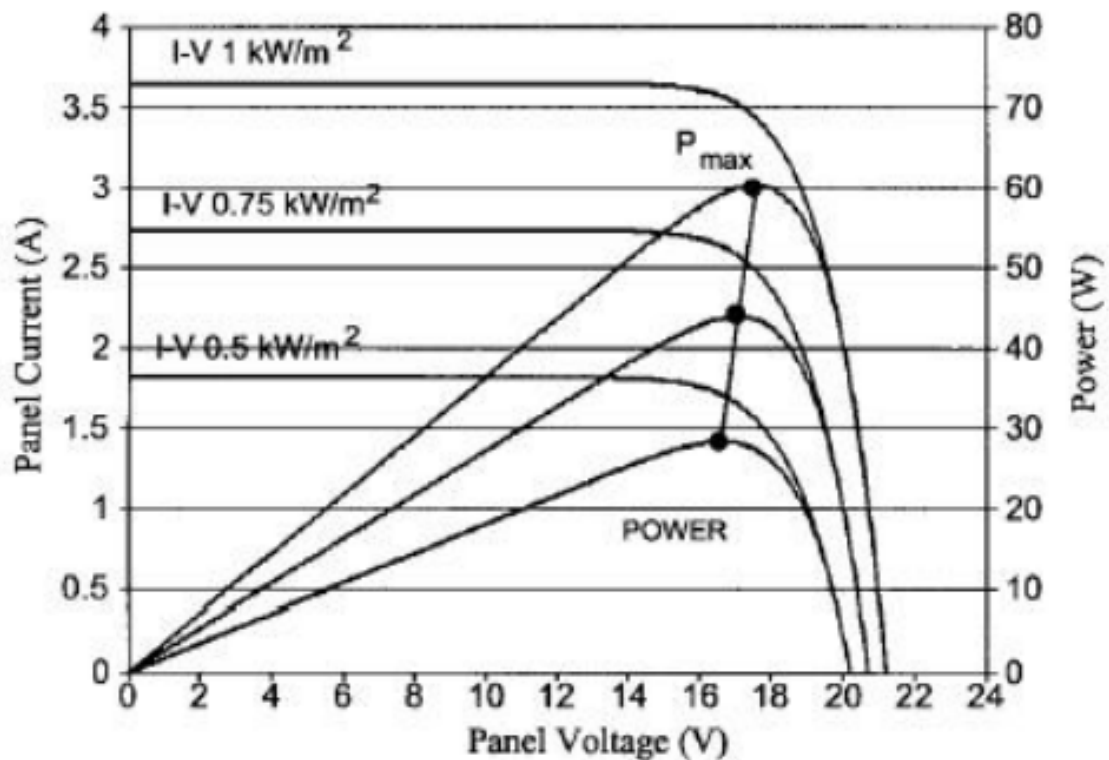


Figura 2.3: Curvas características I-V dependientes de la radiación solar.  
[14]



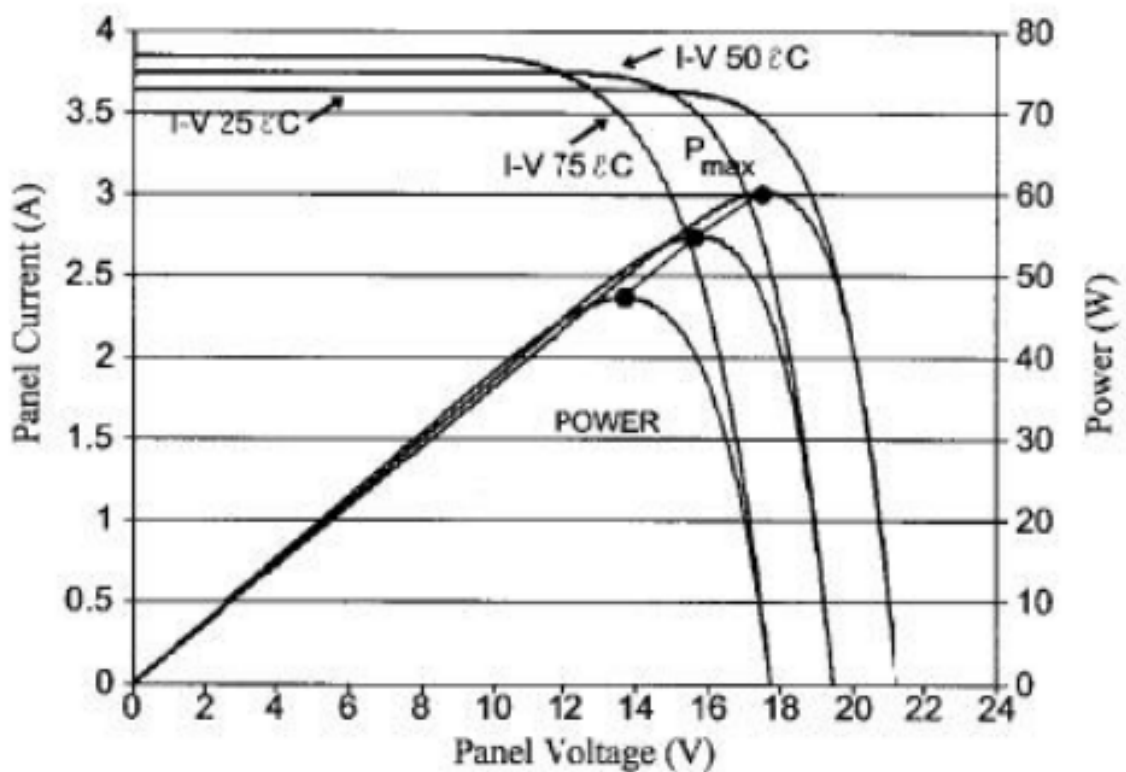


Figura 2.4: Curvas características I-V dependientes de la temperatura.  
[14]

Se observa que a medida que se capte mayor radiación solar, la intensidad generada por el panel aumenta su valor. Por otro lado, la tensión del panel disminuye conforme la temperatura ambiente aumenta.

Los sistemas fotovoltaicos pueden clasificarse como autónomos o conectados a red dependiendo de si estos están o no conectados a la red eléctrica; para efectos de este proyecto de graduación, se enfocará en los sistemas conectados a red los cuales no requieren de baterías para el almacenamiento de energía. En la figura 2.5 se ilustra un esquema básico de un sistema conectado a red:

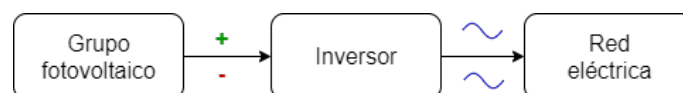


Figura 2.5: Esquema de Sistema Conectado a Red

Los paneles fotovoltaicos al recibir la radiación solar la convierten en energía, pero la electricidad producida es de corriente continua (CC) y hay que recalcar que en la mayoría de los hogares lo que se utiliza es electricidad de corriente alterna (CA). Todas las instalaciones fotovoltaicas, tanto grandes como las pequeñas instalaciones domésticas que requieran convertir la corriente continua en corriente alterna, necesitan un equipo de electrónica de potencia llamado inversor, el cual permite conectar a los paneles con la

red eléctrica y así consumir la energía captada por los paneles fotovoltaicos. El inversor es la interfaz para interconectarlos a las cargas o a la red eléctrica, las cuales funcionan con corriente alterna. [15] Dentro de los tipos de inversores se encuentran:

- **Inversor Central:** El inversor central se basa en la conexión de una gran cantidad de paneles. Estos paneles se conectan en serie dependiendo de la tensión que sea requerida. Los módulos conectados en serie conforman una cadena y esta puede ser replicada para conectarse en paralelo, con el propósito de conseguir mayor potencia. Algunas de las limitaciones que presenta estos inversores son el alto voltaje del cableado de corriente continua entre los módulos y el inversor, pérdidas de potencia debido al seguimiento del punto de máxima potencia centralizado, entre otros (Ver figura 2.6). [16]

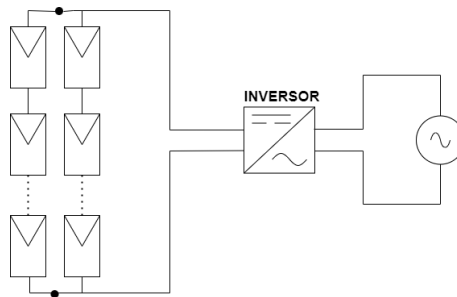


Figura 2.6: Esquema de Inversor Central

- **Inversor de Cadena:** Los inversores de cadena, como su nombre bien lo menciona, se compone de una única cadena de módulos fotovoltaicos que es conectada al inversor. En su entrada puede existir un voltaje lo suficientemente alto como para no utilizar una etapa de la amplificación de tensión. Sin embargo, existe la posibilidad de usar pocos módulos en serie al tener una etapa de amplificación de tensión. Seguidores del punto de máxima potencia pueden aplicarse por separado a cada cadena, aumentando la eficiencia global del sistema comparándolo con el inversor central, y se reduce el precio debido a la producción en serie (Ver figura 2.7). [16]

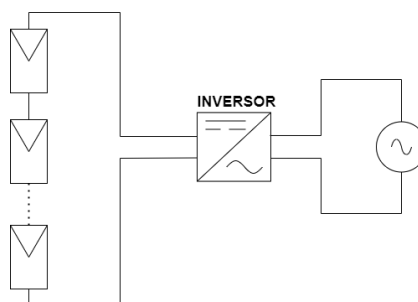


Figura 2.7: Esquema de Inversor de Cadena

Hoy en día los sistemas fotovoltaicos conectados a red (SFCR) se reparten en tres grandes grupos de aplicaciones que son: Instalaciones fotovoltaicas para el sector residencial, en edificios (oficinas, comercios o industria) y plantas fotovoltaicas centralizadas. Cada uno de estos necesita un continuo funcionamiento del sistema de red eléctrica para que actividades diarias no se vean interrumpidas o generen algún daño en dispositivos externos; sin embargo, existen fallas que afectan el funcionamiento efectivo de estos sistemas. Para fines de este proyecto, se tomarán en cuenta la fallas descritas en la tabla 2.1 junto con sus posibles causas y consecuencias. [17] [18]

Falla	Posibles causas	Consecuencias
Circuito abierto	Contracción térmica (ruptura de las interconexiones), instalación ineficiente, vibraciones, uso de componentes no funcionales.	Pérdida de potencia, arcos eléctricos.
Cortocircuito	Instalación ineficiente, deterioramiento del aislamiento del sistema por envejecimiento, humedad, químicos, etc.; cables sueltos en caja de conexiones.	Pérdida de potencia, arcos eléctricos, posible fuego.
Corriente de fuga	Instalación ineficiente, defectos de aislamiento, equipo de baja calidad.	Pérdida de energía, arcos eléctricos (toman años), descargas eléctricas en personas.
Ensuciamiento (Soiling)	Acumulación de suciedad, polvo, nieve u otras partículas que cubren la superficie del módulo fotovoltaico.	Pérdida de potencia y eficiencia.
Sombreado	Desajuste entre celdas, celdas defectuosas, localización no beneficiosa, aves, basura.	Sobrecalentamiento, pérdida de energía, posible destrucción de células o módulo completo.

Tabla 2.1: Tipos de fallas en sistemas fotovoltaicos.

Existe variedad de factores que afectan el rendimiento de un módulo fotovoltaico, tanto la disminución en la irradiancia solar, la temperatura de la celda, y los mencionados en la tabla 2.1. La confiabilidad del sistema general es vital para proporcionar un correcto funcionamiento, de ahí la necesidad de poder rastrear los efectos negativos que traen todos los factores anteriormente mencionados. No solo por el bien del sistema sino también por temas de seguridad, es decir para evitar incidentes cuando se realiza el mantenimiento de la instalación.

En el SESLab se han realizado proyectos con el propósito de mejorar la manera

en la que se obtienen resultados en el estudio de instalaciones fotovoltaicas. Dentro de esos proyectos se encuentran “*Quadratic adjustment with vertex convergence: An alternative algorithm for maximum power point tracking in photovoltaic modules*” con el cual por medio de un algoritmo de seguimiento más suave a los cambios en la radiación solar que ayuda a evitar perturbaciones en la potencia de salida y permite calcular directamente el PMP. [19]

La simulación es una herramienta aliada de los sistemas fotovoltaicos, ya que con esta se puede ahorrar mucho tiempo y dinero en la visualización de problemas. Otros proyectos en donde se han utilizado metodologías de simulación son:

- “*Inverter Configurations Comparative for Residential PV-Grid Connected Systems*” en el cual se simulan tres configuraciones diferentes de inversores fotovoltaicos conectados a la red residencial con el objetivo de realizar una comparativa en términos de restricciones de diseño para estas topologías. [20]
- “*Descripción y simulación de sistemas fotovoltaicos utilizando Modelica y Open Modelica*” donde se desarrolla el módulo de una celda solar combinando, en un sistema simulado, variables eléctricas, físicas y climáticas mientras que en otras plataformas no podrían interactuar. [21]

Todos estos proyectos permiten dar visibilidad a la utilidad que tiene la simulación hoy día, y cómo por medio de ésta se utilizan menos recursos físicos y menos experimentaciones fallidas que al mismo tiempo generan un gasto económico.

## 2.2. Sistemas Causales y Acausales

Cuando se utilizan lenguajes de programación para el modelado de sistemas físicos, existe una clasificación para estos lenguajes dependiendo de la manera en la que las señales de los sistemas se interpretan o relacionan entre sí afectando de una manera u otra al sistema en sí; estos lenguajes se pueden clasificar como causales o acausales. La causalidad se puede representar como la relación que existe entre dos eventos, en este caso la relación entre la entrada del sistema que puede ser una o varias y la salida de este; por lo que en un sistema causal la salida va a responder o depender únicamente de los valores actuales o anteriores de la entrada, definiendo explícitamente la causalidad del sistema. [22] Por ejemplo, si existe una salida  $y(t)$  y se define una función  $f(\cdot)$  la cual va a determinar el valor de la salida dependiendo del valor de la entrada  $x(t)$ , matemáticamente se representaría como:

$$y(t) = f(x(t)) \quad (2.1)$$

Quiere decir que la salida solo depende de valores presentes y/o pasados de la entrada, por lo que no es posible obtener una salida sin que se aplique la entrada. A nivel de programación quiere decir que no se puede llamar a una variable en la salida sin haberla declarado antes, esto provocaría un error y el programa no funcionaría.

Por otro lado, la representación acausal de un sistema es determinada cuando no se definen o especifican las variables de entrada y salida (sentencias de asignación). En lugar de esto se puede recurrir a puertos de interconexión en donde se representa cómo el sistema puede interactuar con otros elementos. Por ejemplo, en lugar de describir un objeto de manera física este puede ser descrito por medio de un algoritmo y ser asignado a otro programa independiente. [21]

Tomando otro escenario de ejemplo para la no causalidad, los elementos pueden ser un sistema (subcircuito) descrito en un script aparte del script principal, el cual es capaz de conectarse e interactuar con otros elementos del sistema. Entonces, cuando el sistema hace la conexión con los demás elementos este puede recibir múltiples variables de entrada que interactúan con otras ya sea por medio de ecuaciones o algoritmos predictivos para generar nuevos valores y al mismo tiempo obtener múltiples variables a la salida. Ya que esto no se denomina explícitamente como un sistema causal, se designa como un sistema acausal.

Se debe entender que para poder obtener a la salida cierto resultado, por ejemplo una gráfica, el script necesita valores de entrada que permitan proporcionar información de lo que se quiere que cierto código realice. Pero también pueden existir variaciones en la entrada a medida que se ejecuta el programa. Como se menciona anteriormente, pueden existir conexiones entre archivos sin conocer aún las características de los elementos que los componen.

## **2.3. Lenguaje de descripción de hardware**

Los lenguajes de descripción de hardware o HDL son lenguajes que permiten describir el funcionamiento y estructura de un circuito eléctrico o digital, ya sea por medio de diagramas esquemáticos donde se muestra la arquitectura del sistema, o de comportamiento al describir el funcionamiento del circuito en lugar de señalar los elementos de los que está compuesto. Los HDL son utilizados para poder describir circuitos en todos los niveles, desde la arquitectura del sistema conceptual hasta los circuitos de bajo nivel. A partir de la descripción del circuito es posible simular su comportamiento mediante herramientas informáticas (simuladores) para comprobar su correcto funcionamiento antes de construir el circuito real. [23]

Dentro de las principales carencias en las herramientas basadas en este lenguaje de simulación actuales, es la necesidad de técnicas de análisis y generación de un diseño

estructurado. Para un mayor provecho de la simulación de circuitos, se pretende trabajar con pequeñas porciones de código que faciliten el desarrollo y se reutilicen bibliotecas previamente utilizadas en códigos anteriores. Con esto, se simplifica las modificaciones constantes dentro del código, de igual forma permite un enfoque al análisis de lo que verdaderamente se desea analizar. [24]

## 2.4. Comparación de herramientas

Para el desarrollo del entorno de simulación, se utilizó el lenguaje de programación *Python*. Una pregunta que puede surgir con esto es, ¿Por qué utilizar *Python* antes que otras herramientas existentes de simulación esquemática de circuitos?, y la respuesta, en términos generales, es debido a la flexibilidad e independencia que se tiene con los componentes que se van a desarrollar; además de la capacidad que se tiene de analizar los datos con los que se interactúa dentro de la simulación. En la tabla 2.2 se muestra una pequeña comparación cualitativa entre algunas herramientas de simulación para ampliar un poco más la justificación de la elección de la herramienta.

Criterio	<i>NI Multisim</i>	<i>LTSpice</i>	<i>Proteus</i>	<i>Eagle</i>	<i>Python</i>
GUI	Si	Si	Si	Si	No
Simulación	Si	Si	Si	No	Si
Funcionamiento en Windows	Si	Si	Si	Si	Si
Funcionamiento en otros sistemas operativos	No	No	No	Si	Si
Código abierto	No	Si	No	Si	Si
Importar archivos de cualquier otra herramienta	No	No	No	No	Si
Análisis complejo con datos de salida	No	No	No	No	Si
Almacenar variables en archivos locales	Si	Si	Si	Si	Si

Tabla 2.2: Comparación cualitativa entre distintas herramientas.

Con la tabla anterior se puede visualizar las ventajas y desventajas que poseen las distintas herramientas de simulación. [25] Ya que se desea trabajar con el mayor dinamismo y flexibilidad posible la herramienta más adecuada para mejorar ciertas limitaciones que tienen las herramientas más utilizadas hoy día para la simulación de circuitos es el lenguaje de programación *Python*.

## 2.5. Python

*Python* es un lenguaje estable, de código abierto y comúnmente utilizado hace varios años atrás. Este se utiliza en diversos proyectos de grandes compañías y de diferentes formas, ya sea como lenguaje de programación básico o para crear extensiones y aplicaciones con un nivel alto de complejidad. Este proyecto se inició en los primeros años de la década de 1990 por Guido Van Rossum en el CWI (Centrum Wiskunde & Informatica) en los Países Bajos. [26]

Para tener claridad con lo que es *Python*, a grandes rasgos este se define como un lenguaje de programación de alto nivel el cual tiene como objetivo de diseño una mayor legibilidad del código y otorgar estructuras permitiendo que programas tanto a pequeña como a gran escala sean claros. Un neologismo frecuente que es utilizado en la comunidad de *Python* es el “pitónico”. A pesar de poseer varios significados en relación con el estilo del programa, describir a un código como pitónico es referirse a que usa bien los modismos de *Python* (es natural o muestra fluidez en el lenguaje) y que se ajusta a la filosofía minimalista de *Python* y enfatizando la legibilidad. Por el contrario, cuando el código es difícil de entender o es una transcripción aproximada de otro lenguaje de programación se denomina no pitónico. [27]

Ampliando el concepto, *Python* es de propósito general que usa un intérprete y admite múltiples paradigmas de programación, incluidos el funcional, imperativo, orientado a objetos o estilos de procedimiento. Es caracterizado por su flexibilidad gracias a la capacidad para utilizar elementos modulares diseñados en otros lenguajes de programación; además cuenta con un sistema de tipo dinámico y una administración automática de memoria. Dentro de las características beneficiosas a tomar en cuenta cuando se utiliza *Python* son las siguientes: [28]

- Lenguaje Poderoso: *Python* tiene una gran biblioteca estándar, considerada como una de las mayores fortalezas de *Python*, que proporciona todas las herramientas adecuadas para la programación, desde las operaciones básicas hasta las funciones avanzadas.
- Lenguaje Compatible: Es compatible con otras tecnologías como C++ o Java. Además, se crearon algunas alternativas y complementos para *Python* como el repositorio oficial de bibliotecas de terceros para *Python* llamado PyPI que ha contribuido con alrededor de 72000 módulos de software.
- Lenguaje Portátil: Linux, Windows, Mac OS, UNIX, Amigo; entre otros son de los sistemas operativos en los que se pueden utilizar los scripts (denominación en inglés) de *Python* y la salida de estos programas es similar en todas estas plataformas.

- Lenguaje Simple: *Python* posee muchas fuentes de aprendizaje como foros, comunidades de interés, tutoriales, etc. Además, la vasta pluralidad de herramientas de terceros provoca que este lenguaje sea agradable y simple de utilizar incentivando a los usuarios a continuar con la aplicación de este.

Todas estas ventajosas características colaboran en el desarrollo del sistema. También generan mayor claridad para quien lo desarrolla y para usuarios externos quienes deseen modificar o hacer uso del programa. En el siguiente capítulo se explicará a detalle el desarrollo del entorno de simulación desarrollo.

Las secciones anteriormente descritas, colaboran a comprender en mayor medida el funcionamiento e importancia que tienen los sistemas fotovoltaicos en la sociedad; así como sus posibles fallas y efectos negativos. Al mismo tiempo se da una vista general acerca del concepto de lenguaje descriptivo de hardware. Se muestra una comparación de las herramientas disponibles hoy día para la simulación de circuitos, y pone en evidencia las ventajas que tiene *Python* ante las otras herramientas a pesar de no poseer una interfaz gráfica de usuario. También se señalan algunas características ventajosas que tiene el software para mostrar el gran potencial que tiene como herramienta de desarrollo.



---

## Arquitectura del sistema de simulación

En el siguiente capítulo, se detallan los distintos elementos que componen el entorno de simulación desarrollado. Se explican ciertas características requeridas por el sistema, así como el uso de las bibliotecas disponibles relacionadas a los sistemas fotovoltaicos dentro de *Python*.

### 3.1. Configuraciones a simular

Tal como se mencionó en la sección 2.1, los sistemas fotovoltaicos que se van a tomar en cuenta son los conectados a red, específicamente las configuraciones de inversor central e inversor de cadena o string. La configuración de inversor central, puede observarse en la figura 3.1, posee un arreglo de paneles conectada a un único inversor de puente completo lo que genera una disminución en la tensión de salida con respecto a la tensión de entrada; esto también es conocido como convertidor reductor o convertidor buck (denominación en inglés). En esta configuración es más difícil localizar algún fallo e incluso se puede decir que es un poco más riesgoso ya que si se presenta alguna falla se puede caer la producción total de energía.

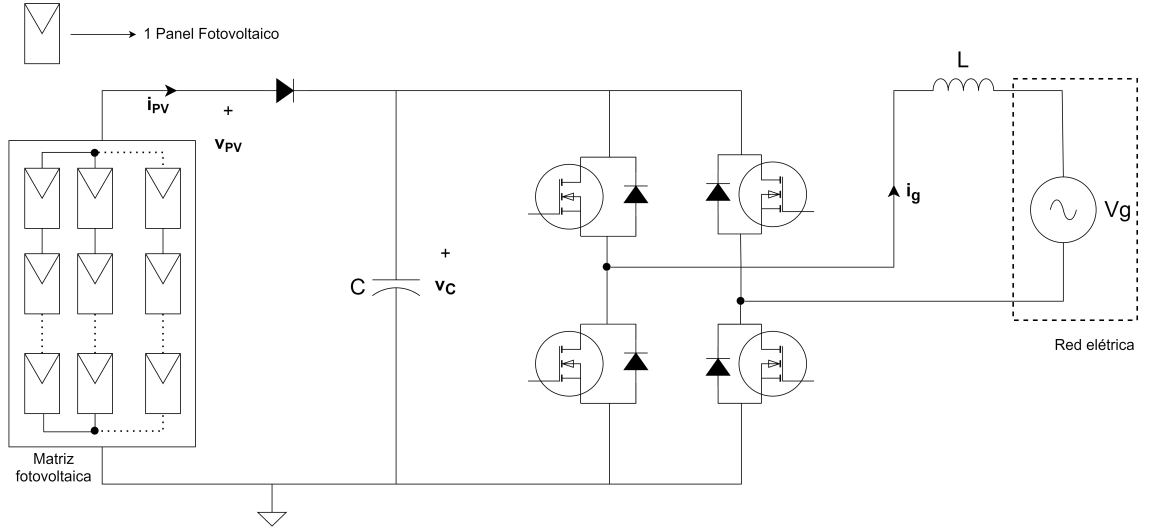


Figura 3.1: Inversor Central

Por otro lado, la configuración de inversor de cadena o string hace alusión a su nombre, ya que está conectada a una única cadena de paneles fotovoltaicos. En este caso, la tensión de salida va a ser mayor que la tensión de entrada gracias al convertidor elevador o Boost (denominación en inglés) de dos etapas el cual se puede visualizar en la figura 3.2.

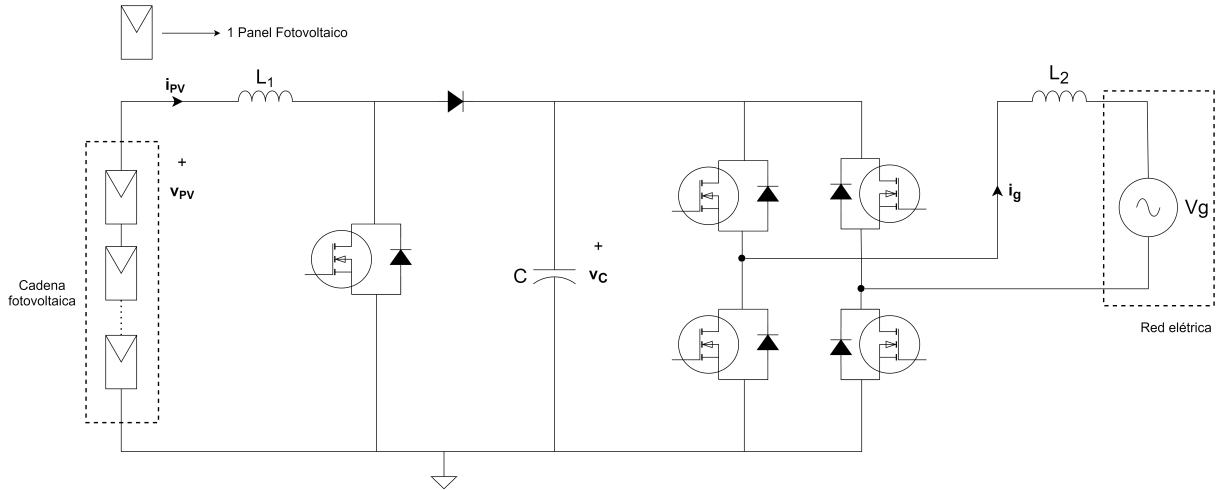


Figura 3.2: Inversor de Cadena

Estas configuraciones de inversores fotovoltaicos presentan una estructura común en cuanto a la etapa CD-CA, ya que ambas disponen de un condensador de enlace CD que llega a actuar como fuente de tensión, que se conecta a la entrada de un convertidor buck de puente completo encargado de realizar la conversión CD-CA.

Para el entorno de simulación a desarrollar, del cual se hablará más adelante, se considera los escenarios:

- Ideal: El cual hace uso de componentes que representan la idealidad de otros. Por ejemplo, el conjunto de paneles se representa como una fuente de tensión CD, los transistores de efecto de campo metal-óxido-semiconductor o bien conocidos como MOSFET se representan como un interruptor y la red eléctrica (fuente de voltaje sinusoidal) como una resistencia.
- Real: Se utilizan los modelos de transistor requerido, el conjunto de paneles como la conexión entre configuraciones equivalentes del panel.
- Fallas: Este último escenario le aplica fallas de cortocircuito, circuito abierto y sombreado/ensuciamiento al escenario real.

## 3.2. Entorno de simulación

Como se mencionó en el capítulo anterior, la herramienta utilizada para el desarrollo del entorno de simulación es *Python*, debido a las grandes ventajas que conlleva su uso.

### 3.2.1. Características del sistema

Es importante mencionar que el entorno de simulación se creó mediante una máquina virtual por lo que en la tabla 3.1 se muestran las características tanto de la máquina física como de la máquina virtual para el desarrollo del entorno.

Características de máquina virtual	
Máquina Virtual utilizada	Oracle VM VirtualBox
Sistema operativo	Ubuntu
Distribución	20.04.2 LTS
Memoria base	4 GB
Procesador (es)	4 CPUs
Características de máquina física	
Memoria RAM	8 GB
Procesador	Intel® Core™ i7-4510U
Sistema Operativo	Windows
Requerimientos del sistema	
<i>Python</i>	Versión 3.8.5
Simulador <i>ngspice</i>	Versión 31

Tabla 3.1: Características del sistema.

La máquina virtual permite no necesitar recursos físicos adicionales en caso de tener poco almacenamiento en el sistema físico. Para fines de este proyecto, por medio de una máquina virtual la instalación de *PySpice* y *Ngspice* es más simple. Es importante

seleccionar adecuadamente la asignación de núcleos de CPU al igual que la memoria RAM cuando se trabaja con una máquina virtual, esto para mejorar el rendimiento de la máquina y evitar ralentizar la generación de simulaciones.

### 3.2.2. Comparación de bibliotecas

Para la utilización de *Python*, es importante inicialmente seleccionar las bibliotecas que pueden ser útiles y definir cuáles se ajustarán o tendrán mayor funcionalidad en el desarrollo del entorno de simulación. Inicialmente se toman en consideración cuatro posibles bibliotecas relacionadas a la simulación de sistemas fotovoltaicos con *Python*, se da una descripción general de estas en la tabla 3.2.

Biblioteca	Descripción
<i>PySpice</i>	Módulo de <i>Python</i> de código abierto que proporciona una interfaz o conecta <i>Python</i> con los simuladores de circuito <i>Ngspice</i> y <i>Xyce</i> . Permite definir un circuito a través de una lista de redes o mejor conocido como netlist.
<i>ltspice</i>	Biblioteca utilizada con la finalidad de interactuar con la herramienta <i>LTSpice</i> . Esta extrae datos de archivos de salida de <i>LTSpice</i> (.raw) y con estos genera y manipula las distintas visualizaciones resultantes para el respectivo análisis.
<i>Spicepy</i>	Biblioteca que nace como un proyecto didáctico, la cual importa lista de redes o netlist que proporcionan la solución del circuito descrito utilizando el análisis nodal modificado.
<i>pulib Python</i>	Herramienta de código abierto que tiene como objetivo proporcionar un conjunto de funciones y clases para simular el rendimiento de los sistemas fotovoltaicos. [29]

Tabla 3.2: Posibles bibliotecas para la simulación de sistemas fotovoltaicos.

Cada una de las bibliotecas poseen características relevantes en relación a los sistemas fotovoltaicos. Sin embargo, las bibliotecas “*ltspice*” y “*Spicepy*” necesitan la importación de archivos que ya posean la descripción de un circuito como tal, tanto de los componentes, sus valores y las conexiones que existen entre estos (archivos .raw, .net o .cir). Con esto, requieren de una segunda herramienta capaz de generar y proporcionar los archivos necesarios para el uso de las bibliotecas anteriormente mencionadas.

Por otro lado, la biblioteca “*pulib Python*”, como bien se menciona, es utilizada en la implementación de modelos relevantes para la energía solar como algoritmos para la irradiancia, posición solar, entre otras, relacionadas más que todo para el análisis del rendimiento de un sistema fotovoltaico empleando el uso de los parámetros. Basándose en la exposición de estas tres bibliotecas y adaptándose a los intereses y objetivos

de este proyecto, se procede a la escogencia de la biblioteca “*PySpice*”. Para realizar la importación se realiza de la siguiente manera en caso de trabajar con el sistema operativo Ubuntu:

```
pip3 install PySpice
```

En caso de utilizar Windows es de la siguiente forma:

```
pip install PySpice
```

La instalación en ambos casos se hacen desde la terminal. Cuando esta se realiza de manera exitosa debería de salir un mensaje en la misma terminal indicándolo.

### 3.2.3. Bibliotecas utilizadas

*PySpice* permite construir desde cero un circuito electrónico, en donde se definen los componentes (tipo, modelo o valor) y las conexiones deseadas entre estos gracias a la conexión con el simulador *ngspice*; además de generar el respectivo netlist. Dentro de sus ventajas se encuentran:

- Ejecución en plataformas como Linux, Windows (64 bits) y Mac OS X.
- Admite simuladores de circuitos *Ngspice* y *Xyce* (En este proyecto se utilizará *Ngspice*).
- Implementa un enlace de biblioteca compartida de *Ngspice* utilizando CFFI que admite fuentes externas.
- Posibilita la implementación de una API de objetos orientados para definir el circuito.
- Trabaja en conjunto con otras bibliotecas para exportar la salida de la simulación.

Al inicio de la sección se plantea la pregunta de, ¿Por qué utilizar *Python* en conjunto con bibliotecas en lugar de un simulador como *KTechlab*, *NI MultiSim* o *LTSpice*? ¿Qué ventajas proporciona *PySpice* a otros simuladores? *Python* es un software libre (código abierto) que puede ser utilizados por cualquier persona para el desarrollo de proyectos. Cuando este se une con la biblioteca *PySpice*, proporciona que se convierta y se use como un language descriptivo de hardware, incluso existen muchos ejemplos en la web que pueden funcionar como guía; así como de una comunidad de la cuál se puede obtener ayuda.

Con *Python* se posee una independencia entre archivos, desde almacenar variables para en un futuro ser analizadas (sin necesidad de ejecutar el programa infinidad de veces), hasta la utilización de módulos provenientes de otros archivos para la

disminución de posibles errores o reducir la complejidad; además, el utilizar este lenguaje de programación asegura efectuar el análisis de datos mediante herramientas modernas. Otra de las ventajas es la velocidad de simulación; cuando el circuito es complejo y posee muchos componentes (en especial si se importan bibliotecas) aplicaciones como *LTSpice* tardan minutos en proporcionar las gráficas resultantes, mientras que *Python* tiene una duración de segundos.

Si bien es cierto, aunque *PySpice* no cuenta con una interfaz gráfica de usuario, también conocida como GUI, este también puede importar netlist desde un editor esquemático como Kicad. Al usar cien por ciento código, se puede describir de forma completa un circuito aunque este tenga una complejidad elevada, y lo conveniente de esto es que se consigue modificar las cosas fácil y rápidamente; por ejemplo, si se desea quitar algún componente este simplemente se comenta, o si se requiere que el valor sea variable tomando en cuenta el interés del usuario, solo se solicita desde la consola.

El entorno de simulación hace uso de otras bibliotecas, a continuación se listarán todas las que utiliza junto con su respectiva funcionalidad y elementos importados en cada una:

→ **PySpice:** Conexión de *Python* con simulador *Ngspice*.

- \* *find\_libraries*: Importa bibliotecas de archivos de *PySpice*.
- \* *plot*: Implementa un ayudante de trazado y parámetros adicionales (waveform, \*args, \*\*kwargs) pasan a la función de Matplotlib. [30]
- \* *SpiceLibrary*: Importa bibliotecas de Spice.
- \* *Circuit*: Permite crear el netlist del circuito.
- \* *Unit*: Importa las unidades de medida.

→ **Matplotlib:** Creación de gráficas estáticas e interactivas en *Python*.

- ◇ *Cursor*: Genera el cursor en las gráficas de salida.

→ **Sys:** Proporciona acceso a variables y hace el llamado de un script de *Python*.

→ **Dill:** Permite almacenar objetos de un script de *Python* en un archivo o cargar estos mismos para ser leídos. El envío de objetos de *Python* se hace a través de la red como un flujo de bytes.

Trabajar en conjunto con las bibliotecas mencionadas anteriormente, permite que el desarrollo del entorno de simulación sea más amigable con el usuario y proporcione mayor entendimiento por si un tercero desea realizar cambios en el programa. Una vez importadas las bibliotecas se procede al desarrollo del circuito.

### 3.2.4. Desarrollo del entorno de simulación

Es recomendable iniciar con la lectura del manual de uso de las bibliotecas, en este caso de *PySpice*, para asegurar que la elaboración se dé de una manera fluida (el manual puede ser encontrado en [30]). Dado que el programa que se va a desarrollar funciona como un lenguaje descriptivo de hardware, no se realiza un diagrama de flujo de este, pero en la figura 3.3 se refleja una serie de pasos generales para cumplir con los objetivos de una simulación exitosa.

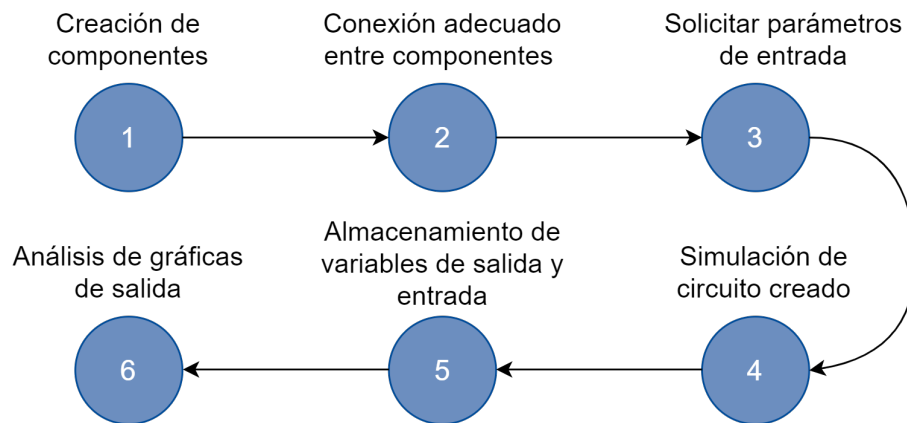


Figura 3.3: Secuencia de pasos a desarrollar

Ahora bien, para el funcionamiento del sistema se requieren tres archivos .py donde cada uno será una pieza fundamental para la ejecución de las simulaciones. Estos archivos componen los casos o escenarios de estudio utilizados que se expondrán más adelante, pero por ahora se explicarán los archivos de forma individual.

- ✓ Archivo utilitario: En este archivo se encuentran las funciones que van a dar utilidad a los otros dos archivos mencionados posteriormente. Estas funciones corresponden a: inicialización del sistema que fusiona las funciones de inicialización de parámetros (componentes y unidades del circuito) e inicialización del simulador (en caso de trabajar con el sistema operativo linux), el almacenamiento de las variables para ser analizadas en un futuro (Se guardan en un archivo .pkl) y la lectura de estas variables. Las bibliotecas utilizadas en el archivo son *PySpice*, *sys* y *dill* (Ver sección 3.2.3).

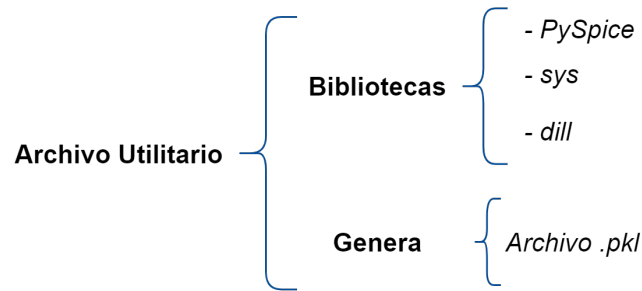


Figura 3.4: Esquema de la composición del archivo utilitario.

- ✓ Archivo generador de gráficas: Como su nombre bien lo menciona, los escenarios se van a componer por un archivo dedicado especialmente a la generación de gráficas. Este puede variar dependiendo del escenario en el que se esté en cuanto a los límites en los ejes "x" y "y", al igual que en las tensiones o corrientes que se quieren visualizar. Este archivo posee las bibliotecas de matplotlib y sys las cuales se comentaron en la sección anterior, al mismo tiempo que las funciones de almacenamiento y lectura de variables. Se debe de recalcar que este archivo puede ser ejecutado dentro del archivo "Creador del circuito" cuando se requiere generar la simulación y obtener los resultados en el mismo instante, o de forma individual cuando se obtienen las variables al ser leídas a partir de un archivo .pkl.

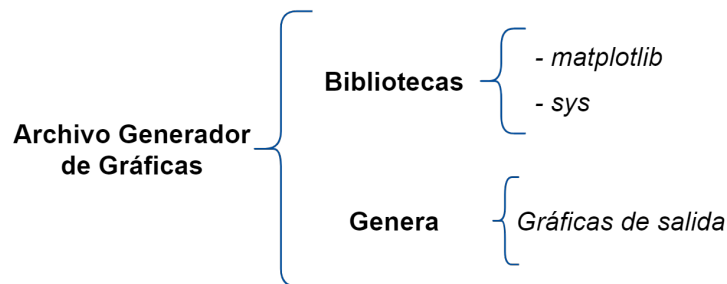


Figura 3.5: Esquema de la composición del archivo generador de gráficas.

- ✓ Archivo creador de circuito: En este script es donde el usuario se encarga de crear el circuito a simular. En el se nombra el circuito, se definen los modelos de componentes como diodos, interruptores o transistores, el tiempo de simulación (en caso de tener una simulación de tipo transitoria), parámetros iniciales del sistema, nombre y conexiones de componentes; en fin, todo lo relacionado a la composición del circuito. Se compone únicamente de la biblioteca *PySpice*, aunque precisa del archivo generador de gráficas y del archivo utilitario, ya que de este solicita las funciones de inicialización del sistema y almacenamiento de variables.



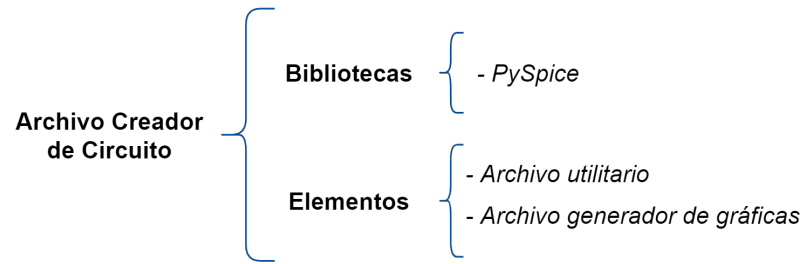


Figura 3.6: Esquema de la composición del archivo creador de circuito.

Para poder modelar el sistema se van a nombrar los archivos a medida de ejemplo como `util.py` (archivo utilitario), `CISI.py` (archivo creador de circuito) y `plotter.py` (archivo generador de gráficas); además, se va a definir el lenguaje de modelado de sistemas de software conocido como "Lenguaje Unificado de Modelado" o UML, así como tipos de diagramas utilizados en este proyecto.

El UML está designado a ser un lenguaje mundial para modelar sistemas y así comprender la funcionalidad de los sistemas que han sido desarrollados en otras partes del mundo. Este lenguaje de modelado dispone de un conjunto amplio de elementos a graficar como componentes, clases, casos de uso, objetos, flujos de trabajo; del mismo modo que la relación entre estos. Dentro de un marco general, su objetivo es describir el comportamiento, estructura, límites y objetos que forman parte del sistema. [31]

En este proyecto, se ilustrará el diagrama de componentes del entorno de simulación (diagrama estructural) y el diagrama de caso de uso (diagrama de comportamiento). [32] El *diagrama de componentes* ilustra las dependencias entre los conjuntos de componentes del sistema de software y la organización entre estos. Con este diagrama se pretende modelar las distintas partes que unidas forman al sistema en sí (figura 3.7).

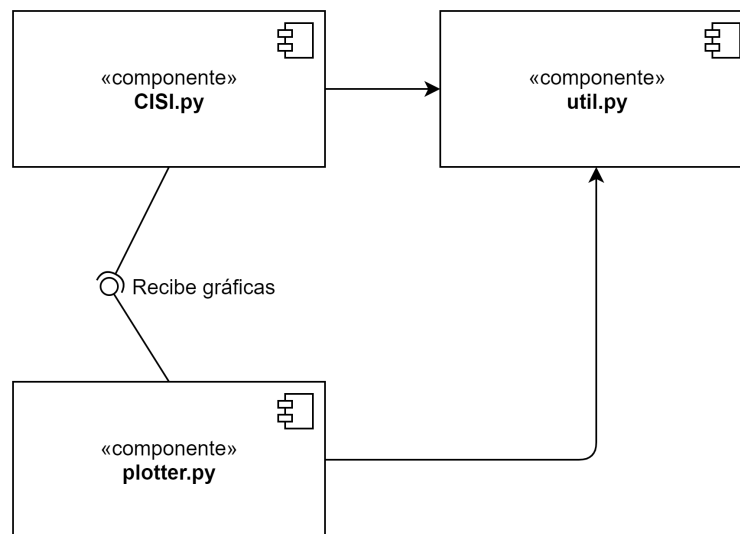


Figura 3.7: Diagrama de Componentes

Por otro lado, el *diagrama de casos de uso* presenta los actores que están involucrados dentro del sistema, las principales funciones que realizan los actores y los principales procesos. Resumiendo, describe las interacciones que se dan entre el actor y el sistema para ejecutar alguna tarea específica. En este tipo de diagramas se presentan relaciones entre casos de uso, las cuáles son de inclusión (include) donde un caso de uso incluye a otro, y extensión (extend) cuando un caso de uso extiende su funcionalidad a otro caso de estudio. Ambos se representan con una flecha de un caso de uso a otro con la etiqueta entre comillas angulares (« »). El diagrama de caso de uso del sistema desarrollado puede verse en la figura 3.8. [33]

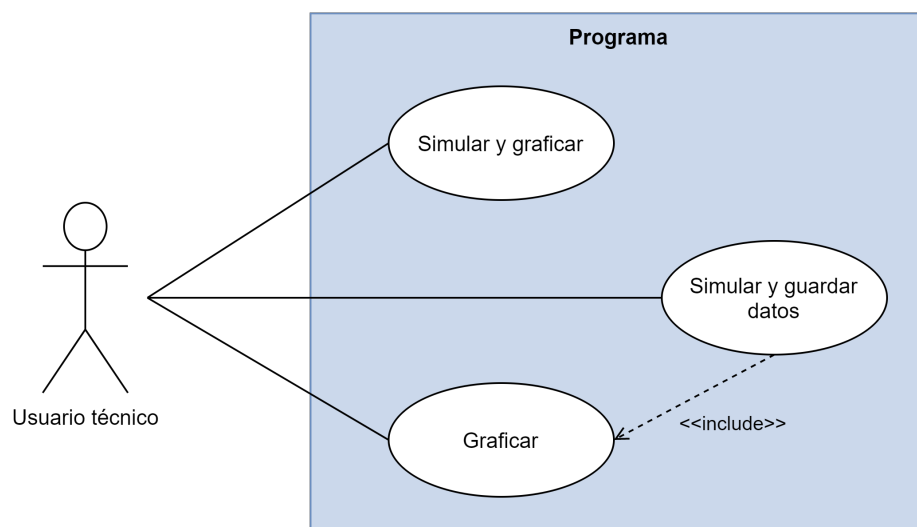


Figura 3.8: Diagrama de Casos de Uso

Es importante destacar que la documentación del desarrollo del sistema genera mayor facilidad en futuros rediseños y al mismo tiempo involucra a los usuarios en las etapas iniciales del sistema. Los diagramas UML permiten que el desarrollador describa el sistema desde su punto de vista, ocasionando con esto un mayor provecho del sistema y haciéndolo aún más entendible para quien lo utilice.

#### 3.2.4.1. Escenarios simulados

Para el desarrollo de las simulaciones se tomó en cuenta la utilización de tres escenarios, tanto para la configuración de inversor central como para la configuración de inversor de cadena.

**Escenario 1:** En ambos casos, el primer escenario describe un circuito ideal en donde el conjunto de paneles se representa como una única fuente de tensión y los transistores MOSFET como interruptores controlados por tensión; el escenario ideal

de la configuración de inversor central y de la configuración de inversor de cadena se encuentran en la figuras 3.9 y 3.10 respectivamente.

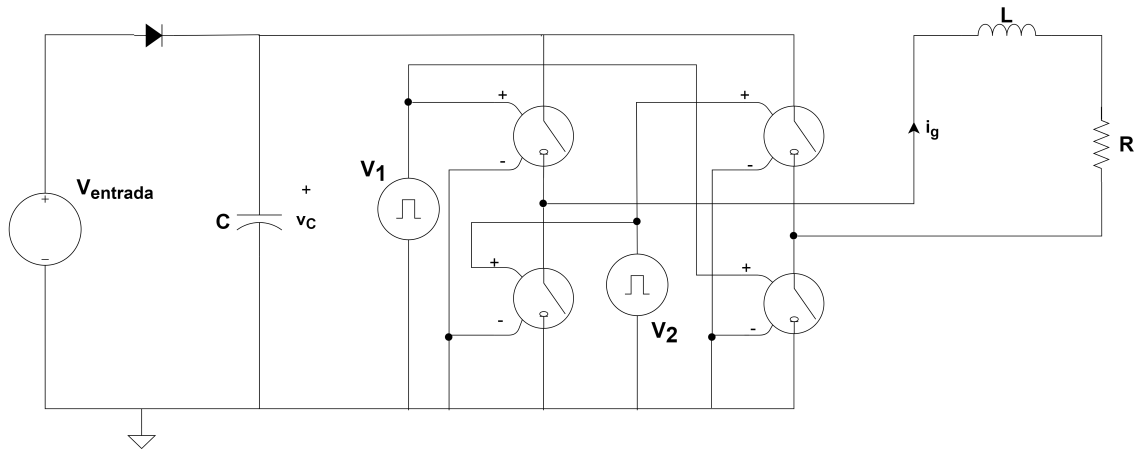


Figura 3.9: Configuración Ideal de Inversor Central

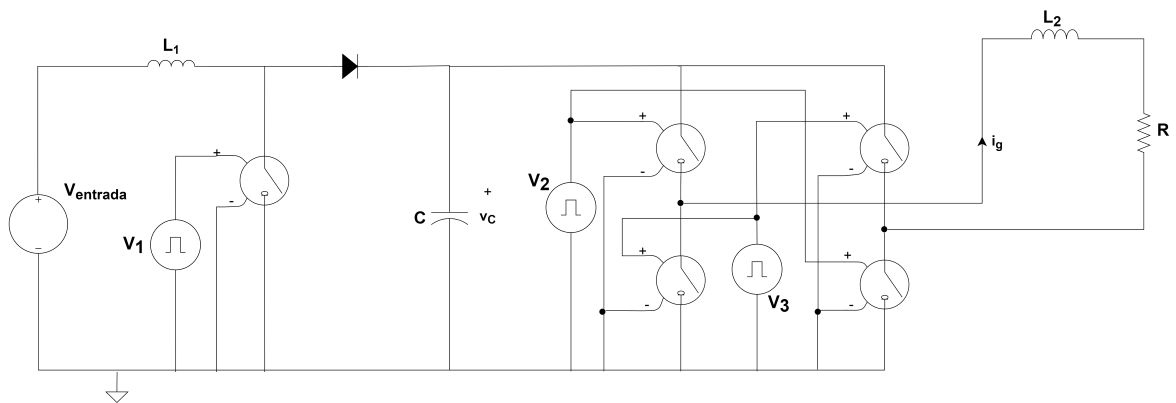


Figura 3.10: Configuración Ideal de Inversor de Cadena

El simular un escenario ideal es equivalente a tomar en cuenta un escenario con cero pérdidas. Por ejemplo, la fuente de tensión que alimenta los circuitos de las figuras 3.9 y 3.10 va a tener un voltaje de salida fijo, independiente de la corriente que consuman los demás componentes conectados a sus terminales.

**Escenario 2:** La diferencia del escenario ideal al escenario real corresponde al cambio de los componentes ideales por componentes reales en el circuito. En este escenario cada panel va a ser simulado como la conexión en paralelo de una fuente de corriente, un diodo y un capacitor (ver figura 3.11). Para el cálculo del valor de la fuente de corriente se toman en cuenta algunos parámetros como la irradiancia, la temperatura, la corriente fotogenerada, entre otros.

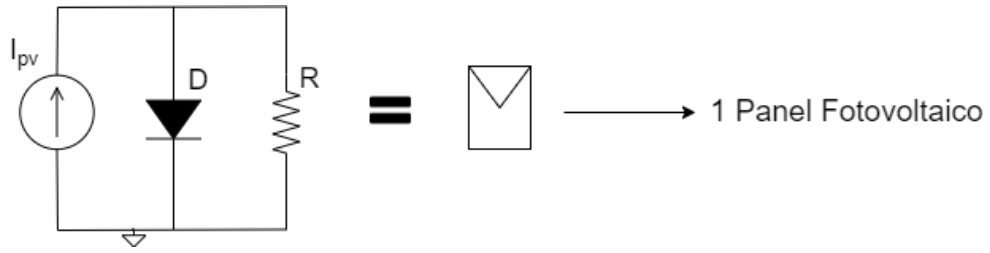


Figura 3.11: Representación de panel fotovoltaico

En las figuras 3.12 y 3.13 pueden ser visualizados las configuraciones de inversor central e inversor de cadena con los componentes reales.

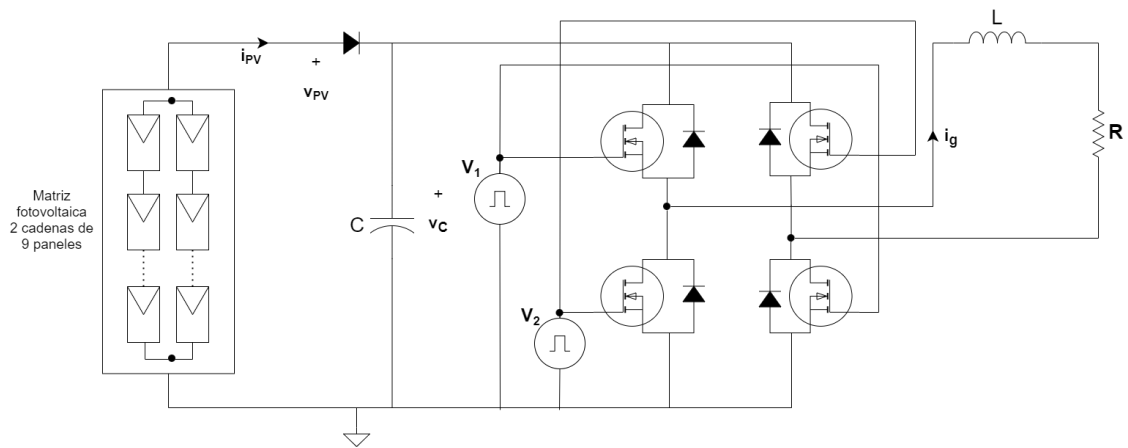


Figura 3.12: Configuración Real de Inversor Central

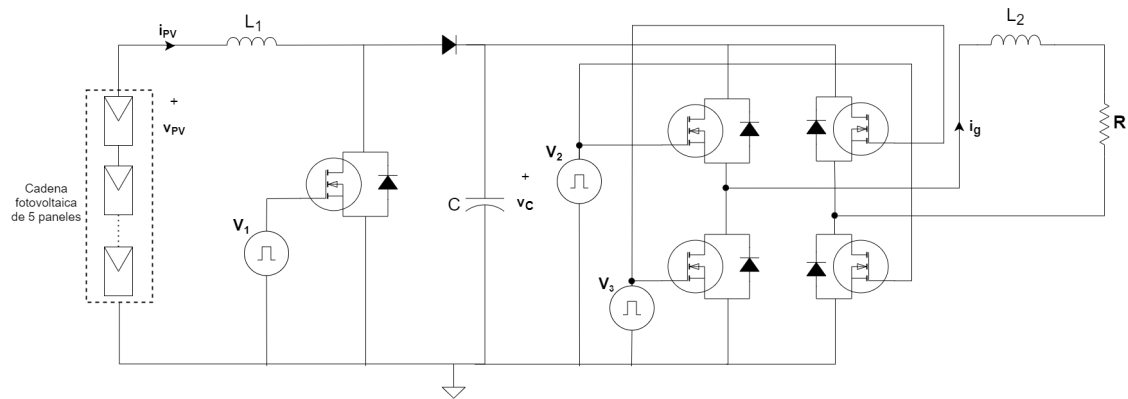


Figura 3.13: Configuración Real de Inversor de Cadena

En este escenario cabe recordar que cuando se utiliza el transistor MOSFET (Canal N) como interruptor se debe de considerar las regiones de corte y saturación. Cuando el voltaje entre la compuerta y la fuente es menor que el voltaje de umbral, el transistor está en la región de corte (No hay conducción). Contrariamente, si el voltaje en la compuerta es suficientemente grande como para que la conexión entre la fuente

y el drenaje se comporte como un corto circuito, el transistor está en la región de saturación.

**Escenario 3:** Como tercer y último escenario, es el correspondiente a la aplicación de fallas eléctricas, de ensuciamiento y sombreado, en el circuito. Las fallas eléctricas aplicadas serán la de cortocircuito, que se representará como un interruptor entre dos cables de un panel para ambas configuraciones (figura 3.14), y la de circuito abierto, que de igual forma se representará como un interruptor (figura 3.15).

En cuanto al sombreado y ensuciamiento, se simulan disminuyendo la irradiancia en los paneles. El sombreado por lo general tiene una mayor irradiancia que el ensuciamiento.

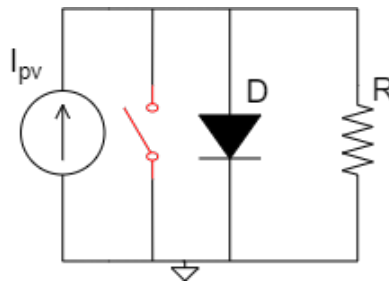


Figura 3.14: Falla de cortocircuito en panel para las configuraciones de Inversor Central e Inversor de Cadena.

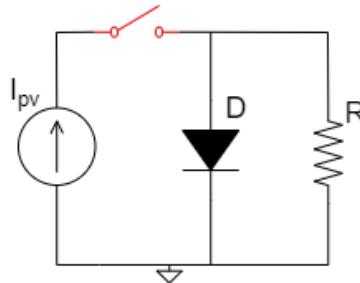


Figura 3.15: Falla de circuito abierto en panel para las configuraciones de Inversor Central e Inversor de Cadena.

Los interruptores que representan la falla de circuito abierto en ambas configuraciones, va a ser controlado por medio de una fuente de voltaje cuadrada para que a la mitad del tiempo de simulación se genere dicha falla en el cable y poder observar las diferencias en la tensión de salida. Esto también se aplica para el interruptor que simula el cortocircuito en el panel.

Cada uno de los escenarios listados anteriormente, posee su respectivo archivo de *Python* para que así las variables generadas en cada uno de estos sean independientes y se puedan almacenar para análisis más complejos o para comparaciones a futuro. Una vez expuesto los elementos que componen el entorno de simulación se procede a mostrar la manera en la que funciona el programa.

### 3.2.4.2. Funcionamiento del programa

Como se mencionó en la sección anterior, no se realiza un diagrama de flujo del programa ya que no corresponde a un algoritmo como tal, debido a que los componentes seleccionados y las conexiones entre estos se van generando a medida que se van declarando. Cabe la posibilidad que el entorno de simulación pueda interactuar con el usuario por medio de la ventana de comandos, como lo que ocurre en esta simulación al permitirle al usuario seleccionar las fuentes de tensión que controlan los interruptores o MOSFET; sin embargo, estas fuentes previamente tuvieron que ser diseñadas dentro del script de *Python*. Para que el entorno de simulación desarrollado funcione adecuadamente y no presente problemas, hay que tomar en cuenta que:

1. Todas las bibliotecas utilizadas deben de estar instaladas correctamente.
2. Todos los archivos que componen el sistema deben de estar en una misma carpeta.
3. Los archivos de *Python* se deben de ejecutar desde la ventana de comandos.

A pesar de no poder generar un diagrama de flujo del entorno de simulación, si se puede representar el uso del sistema mediante este diagrama para brindar mayor claridad en cuanto a la relación que existe entre los archivos que componen el sistema (Ver figura 3.16).

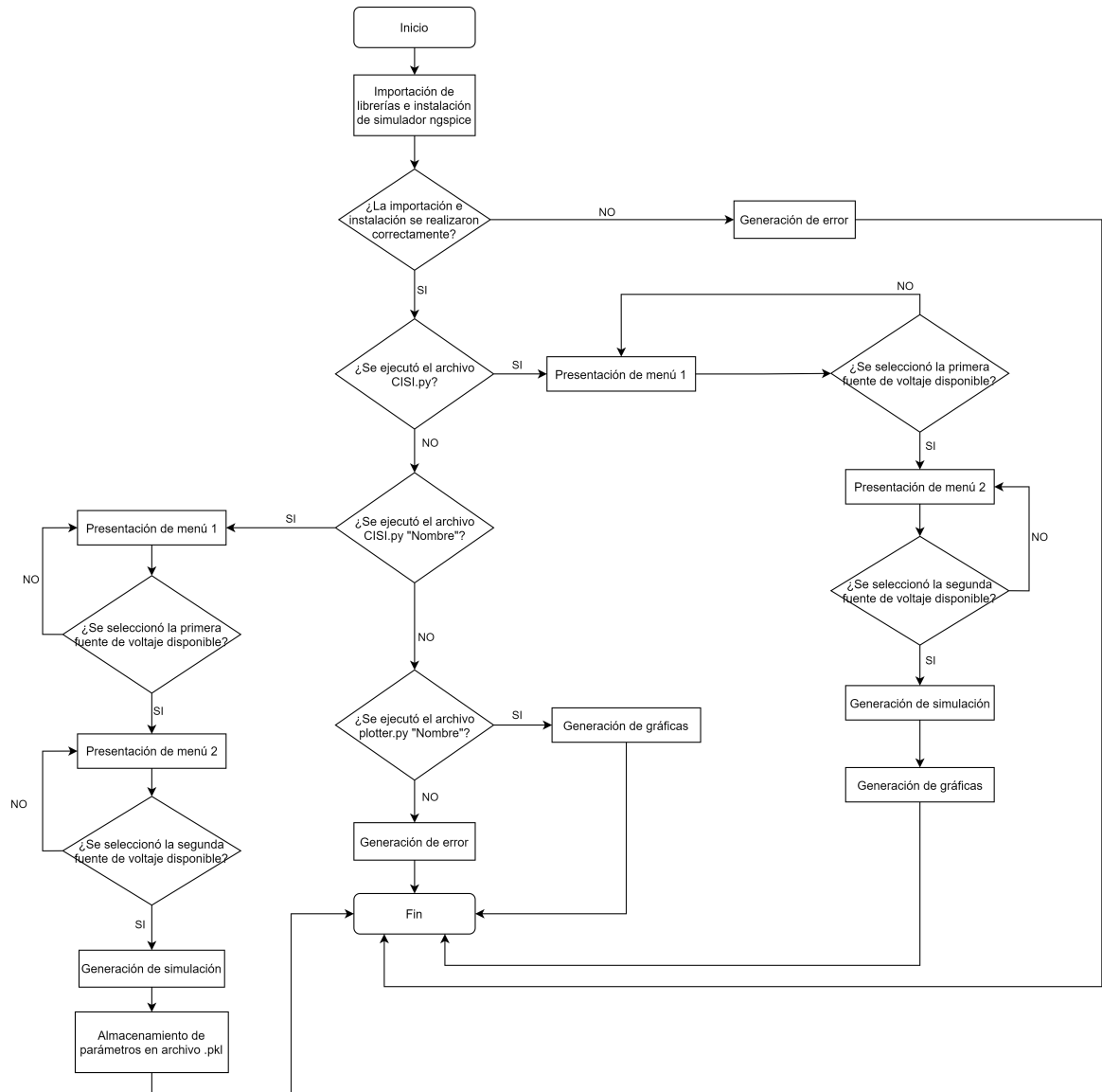


Figura 3.16: Diagrama de flujo de uso del sistema

En las secciones anteriores, se detalla cada uno de los elementos necesarios para el desarrollo del entorno de simulación. En muchas ocasiones puede interpretarse que al existir independencia entre archivos no sean necesarios el uno del otro, argumento que en este caso es erróneo, debido a que cada componente proporciona una funcionalidad esencial para la ejecución del programa. En el siguiente capítulo se facilitan las variables de salida mostradas en gráficas para validar el correcto funcionamiento del entorno, así como el porcentaje de diferencia entre las gráficas generadas por LTSpice y el entorno de simulación.

## Pruebas y verificación del entorno de simulación

Para la verificación del funcionamiento adecuado del entorno de simulación desarrollado, se mostrarán las distintas gráficas de tensión de salida generadas para cada escenario de estudio en las configuraciones de inversor central e inversor de cadena. Además, se adjuntan visualizaciones de la terminal cuando se generan corridas exitosas o errores al ejecutar los distintos archivos .py desarrollados.

### 4.1. Ejecución del programa

Inicialmente, se ejecuta el entorno de simulación desde la terminal:

```
python3 CISI.py
```

Cuando este no posee algún tipo de error, en la ventana de la terminal debe de aparecer un mensaje indicando que el proceso de *PySpice* está iniciando (Ver figura 4.1) y finalmente generar las gráficas respectivas (Ver figura 4.2).

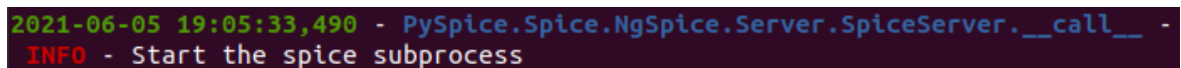
A terminal window with a dark background. The first line is a timestamp and process ID followed by a module path: '2021-06-05 19:05:33,490 - PySpice.Spice.NgSpice.Server.SpiceServer.\_\_call\_\_ -'. The second line is an informational message: 'INFO - Start the spice subprocess'.

Figura 4.1: Ejemplo de ejecución sin fallos del Entorno de Simulación.



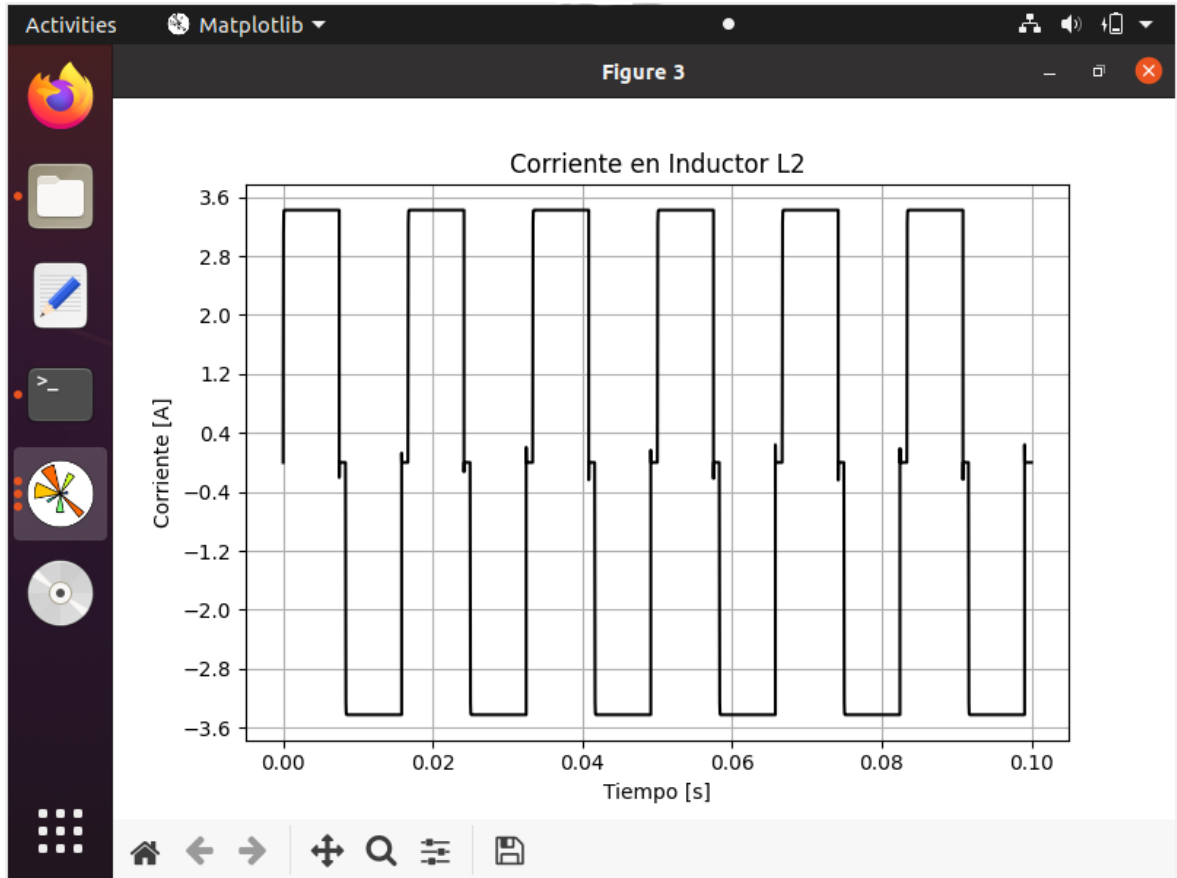


Figura 4.2: Ejemplo de gráfica generada en una ejecución sin fallos del Entorno de Simulación.

Si el entorno de simulación solicita algún tipo de información ; por ejemplo, en este entorno desarrollado se puede hacer la elección de las fuentes que controlan los interruptores. Si se desea obtener las gráficas en el mismo instante en que se ejecuta el archivo creador de circuito (CISI.py), el proceso no iniciará hasta que se haga la selección (Ver figura 4.3) y seguidamente generará las gráficas respectivas (Ver figura 4.2).

```

tania@tania-VirtualBox:~/Desktop/Con menu$ python3 CISI.py
Seleccione la forma de onda para la fuente de voltaje con la que desea controlar el 1er
y 4to interruptor:
    1 - Cuadrada
    2 - Sinusoidal
    3 - Triangular
    4 - Diente de sierra
Inserte la opcion:1

Seleccione la forma de onda para la fuente de voltaje con la que desea controlar el seg
undo y tercer interruptor:
    1 - Cuadrada
    2 - Sinusoidal
    3 - Triangular
    4 - Diente de sierra
Inserte la opcion:1
2021-06-06 01:12:00,228 - PySpice.Spice.NgSpice.Server.SpiceServer.__call__ - INFO - St
art the spice subprocess

```

Figura 4.3: Solicitud de selección de fuentes para el control de interruptores.

Cuando el usuario solo desea almacenar las variables para ser analizadas en un futuro, este deberá de ejecutar el archivo .py en conjunto con el nombre del archivo .pkl en donde se almacenarán dichas variables.

```
python3 CISI.py Prueba
```

Si se da un almacenamiento correcto, en la terminal se muestra lo observado en la figura 4.4:

```

tania@tania-VirtualBox:~/Desktop/Con menu$ python3 CISI.py Prueba1
Seleccione la forma de onda para la fuente de voltaje con la que desea control
ar el 1er y 4to interruptor:
    1 - Cuadrada
    2 - Sinusoidal
    3 - Triangular
    4 - Diente de sierra
Inserte la opcion:1

Seleccione la forma de onda para la fuente de voltaje con la que desea control
ar el segundo y tercer interruptor:
    1 - Cuadrada
    2 - Sinusoidal
    3 - Triangular
    4 - Diente de sierra
Inserte la opcion:1
2021-06-06 12:21:31,898 - PySpice.Spice.NgSpice.Server.SpiceServer.__call__ -
INFO - Start the spice subprocess

```

Figura 4.4: Ejemplo de correcto almacenamiento de variables.

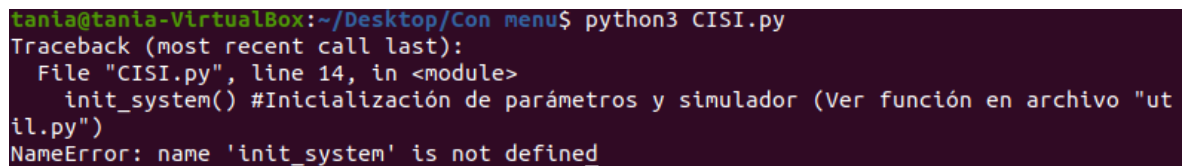
La diferencia que existe entre las ejecuciones de las figuras 4.3 y 4.4 es que en la primera se abrirá una ventana con las gráficas generadas (tomar de referencia la figura 4.2), mientras que en la segunda el proceso se inicia y una vez almacenadas las variables el proceso finaliza automáticamente.

Ahora si el usuario ya posee las variables dentro del archivo .pkl y solo desea graficarlas, simplemente ejecuta el archivo generador de gráficas (plotter.py) junto con

el nombre que posee el archivo .pkl . Una vez ejecutado se generan las respectivas gráficas.

```
python3 plotter.py Prueba
```

Por otro lado, en caso de que exista algún error dentro del programa, la terminal procederá a enviar un aviso con el posible error y la línea de código en donde se encuentra posicionado dicho error. Es necesario asegurarse que los todos los elementos a utilizar (archivos .py) estén dentro de una misma carpeta, para así no generar gráficas erróneas o un error por la no existencia de un archivo requerido. A medida de ejemplo, se generó de forma intencional un error eliminando la declaración del archivo utilitario (util.py) en el archivo fuente (CISI.py). La figura 4.5 muestra la detección del error.



```
tanla@tanla-VirtualBox:~/Desktop/Con menu$ python3 CISI.py
Traceback (most recent call last):
  File "/CISI.py", line 14, in <module>
    init_system() #Iniciación de parámetros y simulador (Ver función en archivo "/util.py")
NameError: name 'init_system' is not defined
```

Figura 4.5: Error en el entorno de simulación

Como se mencionó en la sección anterior, las configuraciones simuladas corresponden al inversor central y al inversor de cadena dentro de distintos escenarios. Para validar el funcionamiento del entorno de simulación desarrollado, se tomará como punto de comparación los resultados obtenidos, simulando estos mismos escenarios, con la herramienta *LTSpice*. Como pruebas se tendrán la gráfica de tensión en la salida del circuito simulado así como la gráfica de la tensión entregada por cada conjunto de paneles.

## 4.2. Cálculo de tiempo promedio de simulación y porcentaje de diferencia entre tensiones

Realizar la validación del entorno de simulación y mostrar las ventajas de este ante la herramienta *LTSpice* se toman en consideración dos variables: la diferencia de tiempo de simulación entre ambas herramientas y el porcentaje de diferencia entre los valores de tensiones obtenidos en los resultados.

El tiempo de simulación es un tiempo aproximado y no exacto ya que en este proyecto se hizo uso tanto de la computadora física como de una máquina virtual. Es importante conocer la diferencia que existe entre el tiempo de simulación de una herramienta a otra por cada escenario simulado, ya que así se demuestra de forma cuantitativa las ventajas que presenta el uso del entorno desarrollado. Esta diferencia de tiempos de simulación se calcula de la siguiente manera:

$$t_{diferencia} = t_{LTSpice} - t_{entornodesarrollado} \quad (4.1)$$

Si se desea obtener un tiempo de diferencia promedio, o bien representar un ahorro de tiempo en la ejecución de las simulaciones de cada configuración mediante una herramienta u otra, se debe de sumar cada uno de los tiempos de diferencia y dividirlos entre la cantidad de tiempos de diferencia calculados:

$$t_{diferencia\ promedio} = \frac{t_{diferencia_1} + t_{diferencia_2} + t_{diferencia_3} + \dots + t_{diferencia_n}}{n\ cantidad\ de\ tiempos\ de\ diferencia} \quad (4.2)$$

Para calcular el porcentaje de diferencia de los valores de tensión en las gráficas de salida entre ambas herramientas, se toma en cuenta el primer valor, que en este caso es el valor de tensión de salida obtenido con la herramienta *LTSpice*, y un segundo valor el cual corresponde al valor de tensión de salida del entorno de simulación desarrollado:

$$\%Diferencia\ entre\ resultados = \frac{V_{LTSpice} - V_{entornodesarrollado}}{V_{entornodesarrollado}} * 100 \quad (4.3)$$

Se puede obtener un porcentaje de diferencia promedio entre los resultados de cada herramienta, y este se calcula de la siguiente forma:

$$\%Diferencia\ promedio\ entre\ resultados = \frac{\%D_1 + \%D_2 + \%D_3 + \dots + \%D_m}{m\ cantidad\ de\ porcentajes\ de\ diferencia} \quad (4.4)$$

Donde:

$\%D$ :  $\%$ Diferencia entre resultados.

La diferencia entre los tiempos de simulación permite observar qué tan ventajoso es la utilización de una herramienta a otra; así como poder calcular un porcentaje de ahorro de tiempo. En cuanto al porcentaje de diferencia entre los valores de tensión de salida, este ayuda a demostrar si verdaderamente en la salida del entorno de simulación se obtienen resultados certeros.

### 4.3. Simulación de la configuración del Inversor Central

Para la configuración de inversor central, el conjunto de paneles fotovoltaicos disponibles están conectados a un único inversor de puente completo en serie-paralelo, y este a su vez tiene conectado en los puntos medios los dos terminales de una carga

compuesta por un inductor y una resistencia en serie. En la salida se espera que la tensión sea menor en comparación con la tensión de entrada, con esto se verifica que el circuito está siendo simulado correctamente por el entorno de simulación.

**Escenario 1:** En este escenario la fuente de tensión que alimenta el circuito es de 350V. Los interruptores van a ser controlados por fuentes de tensión cuadradas, generando la conmutación de dichos interruptores. La figura 4.6a muestra la tensión de salida del circuito en el entorno de simulación desarrollado la cual corresponde a 343 V (Amplitud de onda).

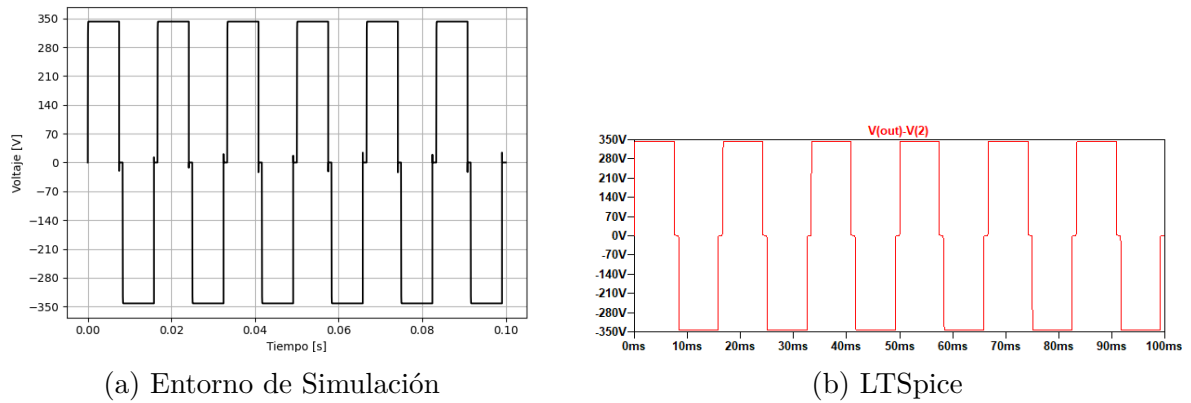


Figura 4.6: Tensión de salida Escenario 1 - Inversor Central

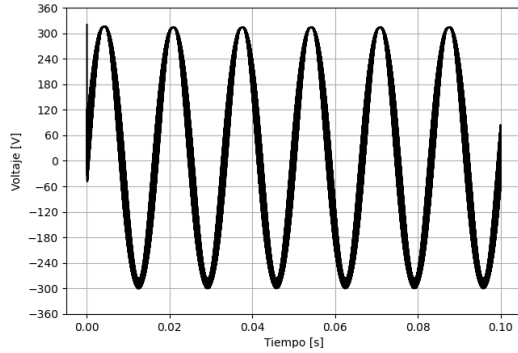
En la figura 4.6 se muestran ambos resultados de la simulación. La tensión de salida en la herramienta de LTSpice es de 342.605V y esto al ser comparado con la obtenida en el entorno de simulación, representa una diferencia de 0.395V que equivale a un porcentaje de diferencia del 0.115 %. En la tabla 4.1 puede observarse un resumen de los resultados.

Inversor Central - Escenario 1		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	342.605	343
$t_{simulación}(s)$	$\sim 240$	$\sim 5$
Diferencia entre resultados (V)	0.395	
%Diferencia entre resultados (%)	0.115	
$t_{diferencia}(s)$	$\sim 235$	

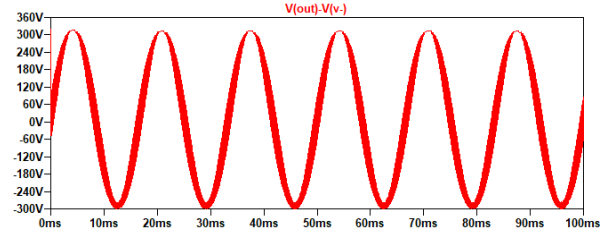
Tabla 4.1: Tabla resumen Inversor Central - Escenario 1

**Escenario 2:** Para este escenario los componentes reales requieren parámetros específicos para funcionar adecuadamente, en este caso hay que considerar las regiones de corte y saturación de los transistores MOSFET canal N, para que así ejecuten la función de interruptor. Para la tensión de entrada se cuenta con dos cadenas de 9 paneles conectados en serie, cada uno de estos paneles genera 40V y en conjunto proporcionan

al circuito una tensión de entrada de 360V. En la figura 4.7a se muestra la tensión de salida obtenida por el circuito.



(a) Entorno de Simulación



(b) LTSpice

Figura 4.7: Tensión de salida Escenario 2 - Inversor Central

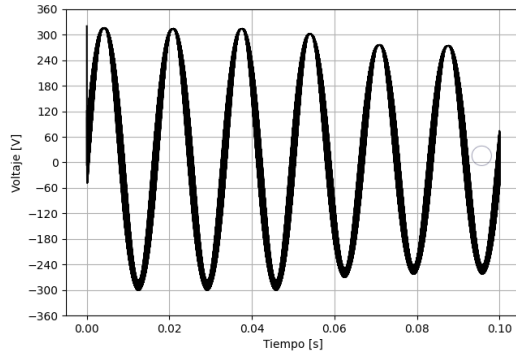
Comparando ambas gráficas, se obtiene que la diferencia de tensiones es de aproximadamente 0.435V, correspondiendo a un porcentaje de 0.138 %. En la tabla 4.2 puede observarse un resumen de los resultados.

Inversor Central - Escenario 2		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	313.565	314
$t_{simulación}(s)$	$\sim 280$	$\sim 7$
Diferencia entre resultados (V)	0.435	
%Diferencia entre resultados (%)	0.138	
$t_{diferencia}(s)$	$\sim 273$	

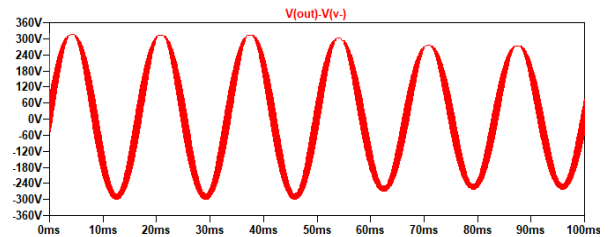
Tabla 4.2: Tabla resumen Inversor Central - Escenario 2

**Escenario 3:** Como último escenario del inversor central, es el correspondiente a la simulación de las fallas en el circuito. A continuación se muestran las gráficas de tensión de salida obtenidas para cada falla y la comparación contra las gráficas generadas en *LTSpice*:

- Cortocircuito: Esta falla eléctrica fue generada en uno de los paneles de la primera de las dos cadenas que conforman el conjunto de paneles de entrada. La falla se colocó para que se generara a la mitad del tiempo de simulación para así poder visualizar con mayor claridad el efecto generado.



(a) Entorno de Simulación



(b) LTSpice

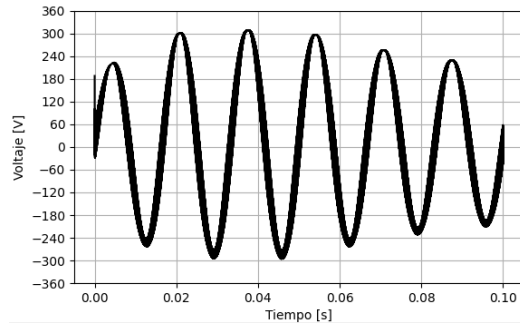
Figura 4.8: Tensión de salida Cortocircuito - Inversor Central

Se puede observar una caída en la tensión a los 50 ms cuando es generado el cortocircuito en el panel. Tomando como referencia la tensión en la última cresta de la onda, en el entorno de simulación (figura 4.8a) la tensión tiene un valor de 273V mientras que en la herramienta de *LTSpice* la tensión es de 272.087V. Así la diferencia entre gráficas tiene un porcentaje de 0.335 %. En la tabla 4.3 puede observarse un resumen de los resultados.

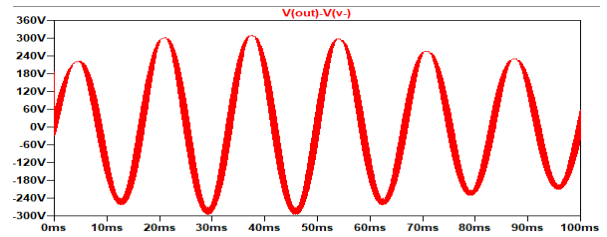
Inversor Central - Escenario 3 Cortocircuito		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	272.087	273
$t_{simulación}(s)$	$\sim 300$	$\sim 10$
Diferencia entre resultados (V)	0.913	
%Diferencia entre resultados (%)	0.335	
$t_{diferencia}(s)$	$\sim 290$	

Tabla 4.3: Tabla resumen Inversor Central - Escenario 3 Cortocircuito

- Circuito abierto: Al igual que el cortocircuito, la falla de circuito abierto fue generada en uno de los paneles de la primera de las dos cadenas que conforman el conjunto de paneles de entrada y se generó a la mitad del tiempo de simulación. En la figura 4.9 pueden visualizarse los resultados.



(a) Entorno de Simulación



(b) LTSpice

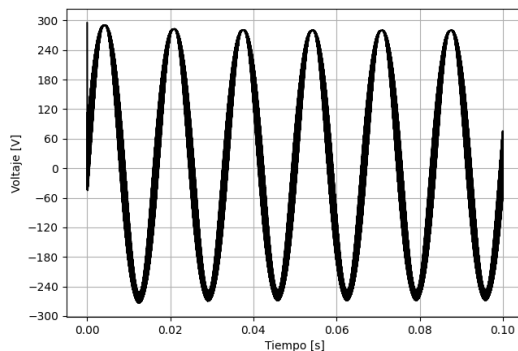
Figura 4.9: Tensión de salida Circuito Abierto - Inversor Central

El circuito abierto produce la desconexión del panel provocando que este no aporte su tensión al resto del circuito. Al igual que en el cortocircuito, la salida a los 50 ms presenta la caída de tensión. La diferencia entre las tensiones de las figuras 4.9a y 4.9b es de 0.855V equivaliendo a un 0.373 %. En la tabla 4.4 puede observarse un resumen de los resultados.

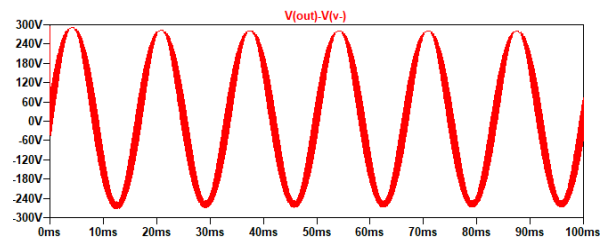
Inversor Central - Escenario 3 Circuito abierto		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	229.145	230
$t_{simulación}(s)$	$\sim 300$	$\sim 10$
Diferencia entre resultados (V)	0.855	
%Diferencia entre resultados (%)	0.373	
$t_{diferencia}(s)$	$\sim 290$	

Tabla 4.4: Tabla resumen Inversor Central - Escenario 3 Circuito abierto

- Sombreado/Ensuciamiento: Ambas fallas se representan en la simulación al disminuir la irradiancia en los paneles. Por este motivo solo se generó una gráfica la cual puede visualizarse en 4.10a.



(a) Entorno de Simulación



(b) LTSpice

Figura 4.10: Tensión de salida Sombreado/Ensuciamiento - Inversor Central



La forma de onda es idéntica que la generada en el escenario 1, con la diferencia que la tensión en la salida es menor. La diferencia que hay entre las tensiones del entorno de simulación y la herramienta *LTSpice* es de 0.695V equivaliendo a un 0.246 %. En la tabla 4.5 puede observarse un resumen de los resultados.

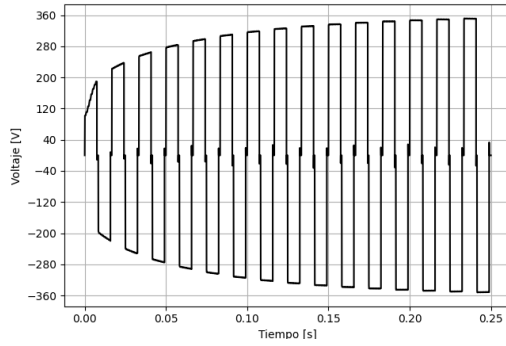
<b>Inversor Central - Escenario 3 Sombreado/Ensuciamiento</b>		
<b>Parámetro</b>	<b>LTSpice</b>	<b>Entorno de simulación</b>
Tensión de salida (V)	282.305	283
$t_{simulación}(s)$	$\sim 240$	$\sim 5$
Diferencia entre resultados (V)	0.695	
%Diferencia entre resultados (%)	0.246	
$t_{diferencia}(s)$	$\sim 235$	

Tabla 4.5: Tabla resumen Inversor Central - Escenario 3 Sombreado/Ensuciamiento

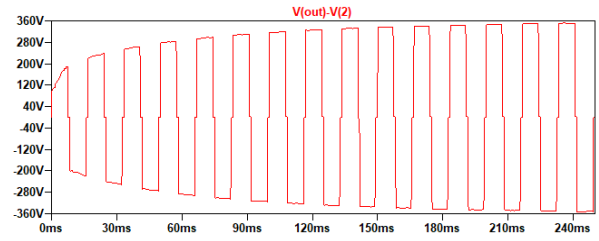
#### 4.4. Simulación de configuración del Inversor de Cadena (String)

La configuración de inversor de cadena, como bien lo menciona su nombre, en la entrada posee un conjunto de módulos fotovoltaicos conectados en serie para formar una cadena. En este circuito, la cadena de paneles está conectada a una etapa de refuerzo de CC o un convertidor Boost-Buck de dos etapas con el fin de inyectar energía fotovoltaica a la red eléctrica. A continuación se presentarán los distintos escenarios simulados con sus respectivas gráficas de salida.

**Escenario 1:** Al ser el escenario ideal, la tensión que alimenta el circuito proviene de una fuente de tensión CD. El valor de esta fuente es de 105V. Al igual que en el inversor central, los interruptores van a ser controlados por fuentes de tensión cuadradas, generando la conmutación de dichos interruptores. La figura 4.11 muestra los resultados de la tensión de salida del circuito en el entorno de simulación desarrollado la cual corresponde a 353 V.



(a) Entorno de Simulación



(b) LTSpice

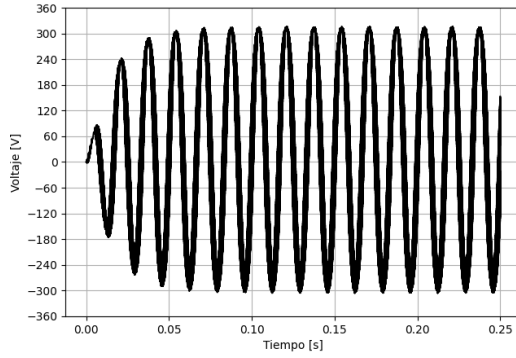
Figura 4.11: Tensión de salida Escenario 1 - Inversor de Cadena

Para la tensión de salida con la herramienta de *LTSpice* se tiene un valor de 351V. Al calcular la diferencia entre ambos valores de salida entre las dos herramientas, se obtiene un porcentaje de diferencia de 0.569%. En la tabla 4.6 puede observarse un resumen de los resultados.

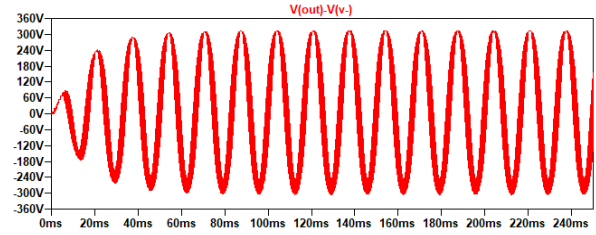
Inversor de Cadena - Escenario 1		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	351	353
$t_{simulación}(s)$	$\sim 360$	$\sim 7$
Diferencia entre resultados (V)	2	
%Diferencia entre resultados (%)	0.569	
$t_{diferencia}(s)$	$\sim 353$	

Tabla 4.6: Tabla resumen Inversor de Cadena - Escenario 1

**Escenario 2:** Este escenario aparte de estar conformado por el puente completo, también lo constituye un MOSFET canal n a la entrada. Dicho transistor va a actuar como un convertidor boost, por lo que se debe de tener cuidado al momento de generar la tensión que se le inyectará en la patilla de la compuerta. Para la tensión de entrada se cuenta con una cadena de 3 paneles conectados en serie. Cada uno de estos paneles genera 40V y en conjunto proporcionan al circuito una tensión de entrada de 120V. En la figura 4.12a se muestra la tensión de salida obtenida por el circuito.



(a) Entorno de Simulación



(b) LTSpice

Figura 4.12: Tensión de salida Escenario 2 - Inversor de Cadena

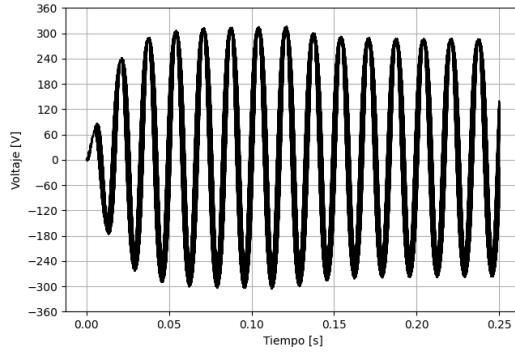
La tensión de salida obtenida en entorno de simulación corresponde a 316V mientras que la obtenida por *LTSpice* es de 314V. La diferencia entre ambos valores de salida es de 2V representando un 0.636 %. En la tabla 4.7 puede observarse un resumen de los resultados.

Inversor de Cadena - Escenario 2		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	314	316
$t_{simulación}(s)$	$\sim 390$	$\sim 10$
Diferencia entre resultados (V)	2	
%Diferencia entre resultados (%)	0.636	
$t_{diferencia}(s)$	$\sim 380$	

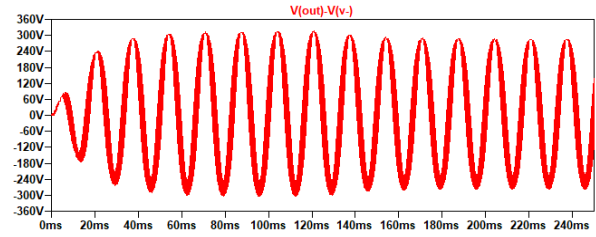
Tabla 4.7: Tabla resumen Inversor de Cadena - Escenario 2

**Escenario 3:** Para finalizar con las pruebas de funcionamiento, se tiene como escenario final la simulación de fallas en el circuito. A continuación se muestran las gráficas de tensión de salida obtenidas para cada falla y la comparación contra las gráficas generadas en *LTSpice*:

- Cortocircuito: La falla eléctrica de cortocircuito fue generada en uno de los paneles de la cadena de paneles de entrada. El efecto que genera esta falla puede observarse en la figura 4.13a después de los 125ms (mitad del tiempo de duración de la simulación).



(a) Entorno de Simulación



(b) LTSpice

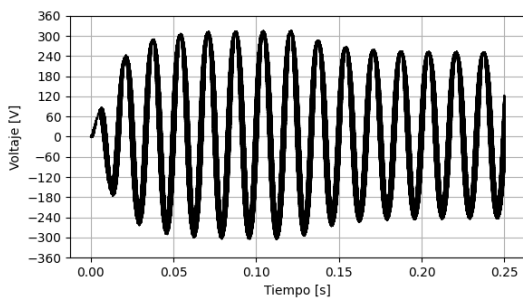
Figura 4.13: Tensión de salida Cortocircuito - Inversor de Cadena

Puede observar como se da la disminución en la tensión de salida de 315.968V a 283.240V al generar un corto circuito en uno de los paneles. Si se comparan los resultados del entorno simulado contra los valores resultantes de la gráfica producida en *LTSpice* se obtiene un porcentaje de diferencia del 0.268 %. En la tabla 4.8 puede observarse un resumen de los resultados.

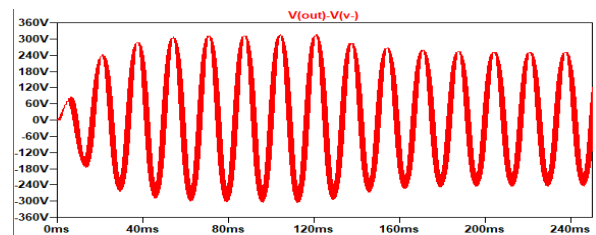
Inversor de Cadena - Escenario 3 Cortocircuito		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	283.240	284
$t_{simulación}(s)$	$\sim 420$	$\sim 15$
Diferencia entre resultados (V)	0.76	
%Diferencia entre resultados (%)	0.268	
$t_{diferencia}(s)$	$\sim 405$	

Tabla 4.8: Tabla resumen Inversor de Cadena - Escenario 3 Cortocircuito

- Circuito abierto: Al generar un circuito abierto en uno de los paneles, se interrumpe la entrega de tensión de este panel al resto del circuito. El resultado obtenido por el entorno de simulación se encuentra en la figura 4.14a.



(a) Entorno de Simulación



(b) LTSpice

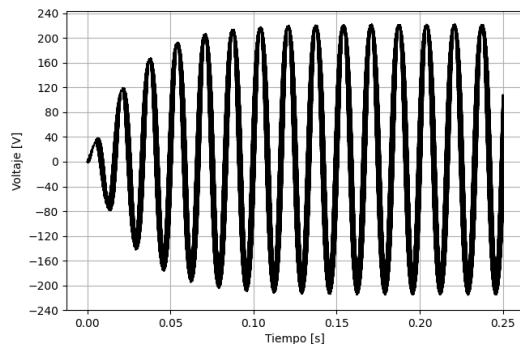
Figura 4.14: Tensión de salida Circuito Abierto - Inversor de Cadena

Al igual que en el cortocircuito, se evidencia una disminución de tensión cuando se aplica el circuito abierto. La diferencia entre las tensiones de las figuras 4.14a y 4.14b es de 1.087V equivaliendo a un 0.348 %. En la tabla 4.9 puede observarse un resumen de los resultados.

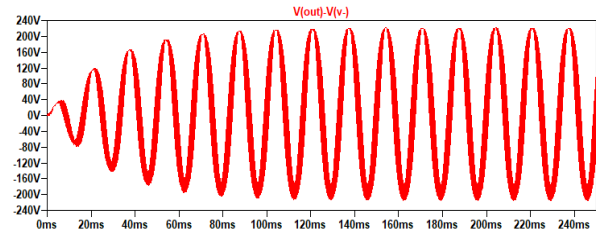
Inversor de Cadena - Escenario 3 Circuito abierto		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	311.913	313
$t_{simulación}(s)$	$\sim 420$	$\sim 15$
Diferencia entre resultados (V)	1.087	
%Diferencia entre resultados (%)	0.348	
$t_{diferencia}(s)$	$\sim 405$	

Tabla 4.9: Tabla resumen Inversor de Cadena - Escenario 3 Circuito abierto

- Sombreado/Ensuciamiento: El sombreado y ensuciamiento generan una disminución de la irradiancia en los paneles. Al disminuir esta se obtuvo el la gráfica de tensión de la figura 4.15a.



(a) Entorno de Simulación



(b) LTSpice

Figura 4.15: Tensión de salida Sombreado/Ensuciamiento - Inversor de Cadena

La tensión en la salida es menor que si se compara con la tensión de salida obtenida en el primer escenario. La diferencia que hay entre las tensiones del entorno de simulación y la herramienta *LTSpice* es de 1.25V equivaliendo a un 0.566 %. En la tabla 4.10 puede observarse un resumen de los resultados.

Inversor de Cadena - Escenario 3 Sombreado/Ensuciamiento		
Parámetro	LTSpice	Entorno de simulación
Tensión de salida (V)	220.75	222
$t_{simulación}(s)$	$\sim 360$	$\sim 7$
Diferencia entre resultados (V)	1.25	
%Diferencia entre resultados (%)	0.566	
$t_{diferencia}(s)$	$\sim 353$	

Tabla 4.10: Tabla resumen Inversor de Cadena - Escenario 3 Sombreado/Ensuciamiento

## 4.5. Síntesis

El entorno de simulación toma una cantidad mínima de recursos computacionales, ya que si se desea, los resultados generados por este pueden ser desechados al momento de finalizar con la simulación; pero al mismo tiempo permite que el usuario decida si quiere almacenar las variables para futuros análisis. Además, posibilita que el usuario intercambie componentes desde la terminal sin necesidad de acceder directamente al archivo fuente.

Este entorno desarrollado al hacer uso de herramientas externas como lo es el caso de la utilización del simulador *ngspice*, facilita el proceso de generación de gráficas y obtiene resultados con alto grado de fiabilidad. Esto puede comprobarse al simular ambas configuraciones de inversores, en donde generando las respectivas gráficas para cada uno de los escenarios de estudio, se demostró que la diferencia de estos resultados contra los resultados obtenidos con la herramienta de simulación *LTSpice* es muy baja. Las figuras 4.16 y 4.17 muestran un gráfico de barras donde se puede apreciar la diferencia en los valores de tensión resultantes en las herramientas utilizadas por los escenarios de cada configuración.

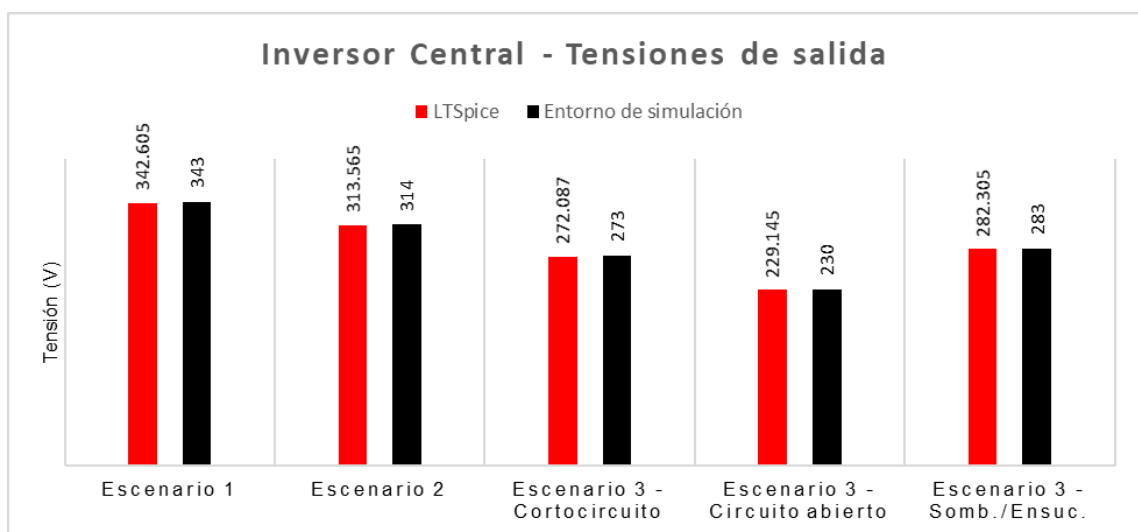


Figura 4.16: Inversor Central - Tensiones de salida en cada escenario de estudio.

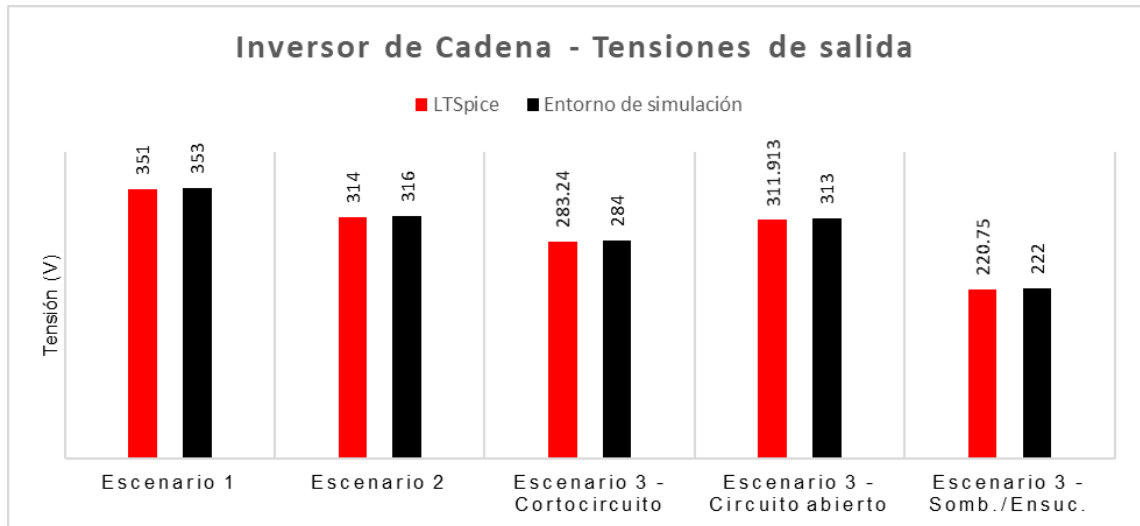


Figura 4.17: Inversor de Cadena - Tensiones de salida en cada escenario de estudio.

Para cada escenario de simulación se calculó el porcentaje de diferencia entre los valores de tensión obtenidos en la herramienta *LTSpice* y en el entorno de simulación. Con estos datos, se puede hacer el cálculo del porcentaje de diferencia promedio que existe entre todos los resultados obtenidos. Por medio de la ecuación 4.4 se obtiene un porcentaje de diferencia promedio:

$$\begin{aligned} \%Diferencia\ promedio &= \frac{0,115 + 0,138 + 0,335 + 0,373 + 0,246 + 0,569 + 0,636 + 0,268 + 0,348 + 0,566}{10} [\%] \\ &= 0,3594\% \end{aligned} \quad (4.5)$$

El resultado anterior, valida que las gráficas obtenidas por el entorno de simulación son acertadas gracias a que este es muy bajo, ya que ni siquiera alcanza el 1 %.

Una de las principales ventajas que presenta el entorno de simulación ante otras herramientas de simulación esquemáticas de circuitos como lo es *LTSpice*, es la velocidad de simulación. Es de relevancia recalcar que para este proyecto se utilizó una máquina virtual, en donde los procesos que se ejecutan dentro de esta se encuentran limitados por los recursos que puede proporcionarle cada usuario. En las figuras 4.18 y 4.19, se muestran los tiempos de simulación de cada configuración por escenario de estudio.

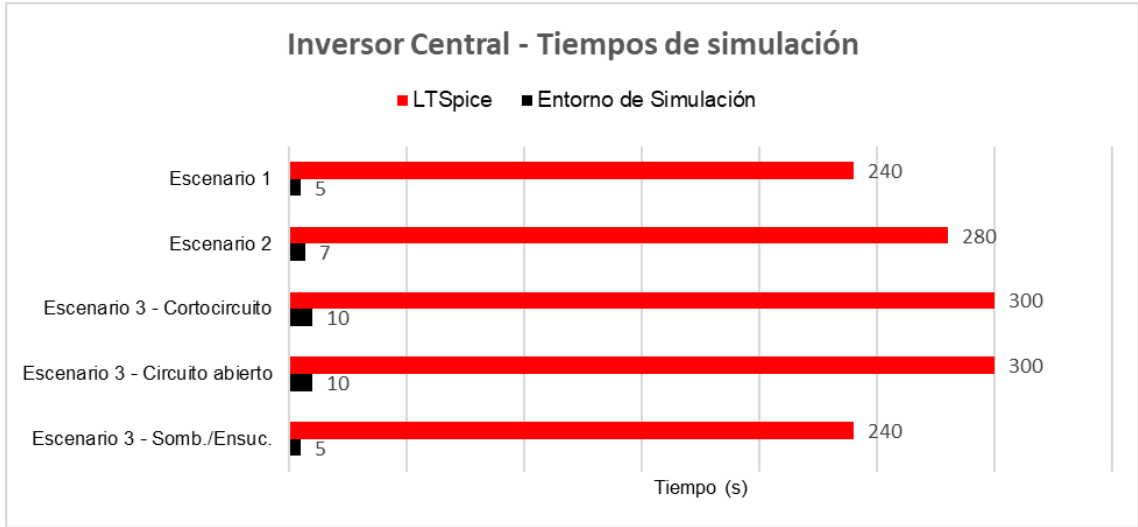


Figura 4.18: Inversor Central - Tiempos de simulación en cada escenario de estudio.

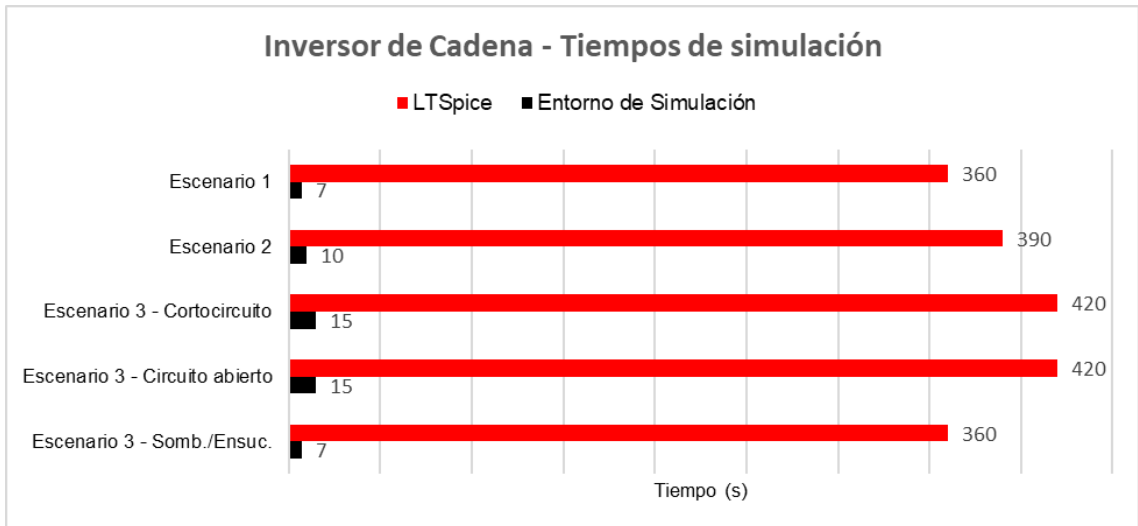


Figura 4.19: Inversor de Cadena - Tiempos de simulación en cada escenario de estudio.

A pesar de trabajar en una máquina virtual, en donde por lo general los procesos se dan de una manera más lenta que en una computadora física, el tiempo de generación de las simulaciones oscila entre los 5 y 15 segundos, mientras que las simulaciones generadas con *LTSpice* tomaron alrededor de 4-7 minutos. Con la ecuación 4.2, se puede obtener el tiempo de diferencia promedio que existe entre todos los resultados obtenidos en el entorno de simulación desarrollado y la herramienta *LTSpice*:

$$\begin{aligned}
 t_{\text{diferencia promedio}} &= \frac{235 + 273 + 290 + 290 + 235 + 353 + 380 + 405 + 405 + 353}{10} [s] \\
 &= 321,9 \text{ s}
 \end{aligned}
 \tag{4.6}$$

Este resultado es equivalente a decir que las simulaciones realizadas con *LTSpice*



toman alrededor de 5.3 minutos más que realizándolas en el entorno desarrollado. Esto demuestra que el usuario con el entorno de simulación desarrollado puede generar resultados en un tiempo considerable, obtener mayor cantidad de gráficas por minuto y con la posibilidad de generar análisis más complejos dentro del mismo programa.

---

## Conclusiones y recomendaciones

### 5.1. Conclusiones

- Se diseñó un entorno de simulación por medio de la biblioteca PySpice la cual proporciona una conexión con el simulador *ngspice* y permite alterar los componentes, conexiones entre componentes, así como los parámetros de estos de acuerdo a la configuración que el usuario desea realizar.
- Se logró emular tres escenarios: ideal, real y con fallas (cortocircuito, circuito abierto y sombreado/ensuciamiento); en las configuraciones de inversor central e inversor de cadena con un tiempo de simulación que oscila entre los 5 y 15 segundos, generando así un ahorro de tiempo aproximado de 5.3 minutos por simulación en comparación con la herramienta *LTSpice*.
- Se consiguió un porcentaje de diferencia promedio de 0.3652% entre los resultados obtenidos con el entorno de simulación y la herramienta *LTSpice*, demostrando la fiabilidad de los resultados obtenidos en el entorno desarrollado.

### 5.2. Recomendaciones

- Para un mayor aprovechamiento del entorno de simulación es recomendable utilizar más bibliotecas disponibles las cuales pueden facilitar el proceso de simulación y crear análisis más complejos con los datos obtenidos en los resultados.
- Ya que el programa posibilita la conexión con otros tipos de archivos como *.pkl*, *.py*, *.txt*, entre otros; hacer uso o importar archivos que contienen configuraciones realizadas por otros usuarios en distintas aplicaciones y así generar un mayor dinamismo dentro del entorno de simulación.

- Una mejora en el diseño del entorno de simulación es implementar una lógica que permita seleccionar la cantidad de paneles en serie o paralelo que se desea por cadena, al igual que la cantidad de cadenas de paneles fotovoltaicos.

---

## Bibliografía

- [1] M. Delgado and F. Calvo. (2017) Sistemas fotovoltaicos, nuevas herramientas de energía renovable. [Online]. Available: <https://www.inteco.org/blog/nuestro-blog-1/post/sistemas-fotovoltaicos-125>
- [2] Grupo Editorial Editec SPA, “Tendencias y desafíos para los inversores,” in *Electricidad*, Abril 2016. [Online]. Available: <https://www.revistaei.cl/reportajes/tendencias-y-desafios-para-los-inversores/>
- [3] C. Meza. (2013, Febrero 5) Seslab promueve e investiga en energías renovables. [Online]. Available: [https://revistas.tec.ac.cr/index.php/investiga\\_tec/article/view/705/636](https://revistas.tec.ac.cr/index.php/investiga_tec/article/view/705/636)
- [4] G. Jiménez, “Nuevo complejo solar producirá del 25 al 30 % de la energía eléctrica del campus tecnológico central,” *Hoy en el TEC*, Abril 26, 2019. [Online]. Available: <https://www.tec.ac.cr/hoyeneltec/2019/04/26/nuevo-complejo-solar-producira-25-30-energia-electrica-campus-tecnologico-central>
- [5] T. Heggarty, “Global solar installations to reach record high in 2019,” *gtm*, Julio 2019. [Online]. Available: <https://www.greentechmedia.com/articles/read/global-solar-pv-installations-to-reach-record-high-in-2019#:~:text=According%20to%20Wood%20Mackenzie%20Power,year%20by%20the%20early%202020s>
- [6] “Future of solar photovoltaic: Deployment, investment, technology, grid integration and socio-economic aspects (a global energy transformation: paper),” International Renewable Energy Agency (IRENA), Abu Dhabi, 2019.
- [7] C. Español, “Dubai está construyendo el complejo de energía solar más grande del mundo,” *CNN*, Abril 2019. [Online]. Available: <https://cnnespanol.cnn.com/video/dubai-energia-solar-complejo-renovable-limpia-planeta-mundo-pkg-digital-orig/>

- [8] A. V. Campos, “Concluye entrega de 296 paneles solares a familias en territorios indígenas de la zona sur,” *Acontecer Digital*, Marzo 2021. [Online]. Available: <https://www.uned.ac.cr/acontecer/a-diario/gestion-universitaria/4312-concluye-entrega-de-296-paneles-solares-a-familias-en-territorios-indigenas-de-la-zona-sur>
- [9] M. R. Maghami, H. Hizam, C. Gomes, M. A. Radzi, M. I. Rezadad, and S. Hajjhorbani, “Power loss due to soiling on solar panel: A review,” *Renewable and Sustainable Energy Reviews*, vol. 59, pp. 1307–1316, 2016.
- [10] T. Tsoutsos, Z. Gkouskos, and S. Tournaki, “Formacion de instaladores fotovoltaicos,” Univ. Técnica de Creta, Grecia, Tech. Rep. IEE/09/928/SI2.558379, 2011.
- [11] D. Moser, M. Del Buono, U. Jahn, M. Herz, M. Richter, and K. De Brabandere, “Identification of technical risks in the photovoltaic value chain and quantification of the economic impact,” *Progress in Photovoltaics: Research and Applications*, vol. 25, no. 7, pp. 592–604, 2017.
- [12] M. Abella, “Sistemas fotovoltaicos,” CIEMAT, Madrid, España, Tech. Rep.
- [13] P. Asensio Gómez, “Combinación de energía solar fotovoltaica y energía minieólica como alternativas limpias para el parcial abastecimiento de una vivienda,” 2020.
- [14] M. Taherbaneh, A. H. Rezaie, S. Karbasian *et al.*, “Combination of fuzzy-based maximum power point tracker and sun tracker for deployable solar panels in photovoltaic systems,” in *2007 IEEE International Fuzzy Systems Conference*. IEEE, 2007, pp. 1–6.
- [15] U. POTOSINOS, “Energía solar fotovoltaica,” *Síguenos:@ revupotosinos Universitarios Potosinos*, p. 25, 2019.
- [16] C. R. Sánchez Reinoso, D. H. Milone, and R. H. Buitrago, “Estudio de inversores fotovoltaicos,” 2010.
- [17] J. Cepeda and A. Sierra, “Aspectos que afectan la eficiencia en los paneles fotovoltaicos y sus potenciales soluciones,” Universidad Santo Tomás, Bogotá, Colombia, Tech. Rep., 2017. [Online]. Available: <https://repository.usta.edu.co/handle/11634/4196>
- [18] J. Garrido, “Estudio de la influencia del ensuciamiento por polvo en la temperatura de un panel solar fotovoltaico mediante un modelo computacional unidimensional,” B.S. Thesis, Universidad de Chile, Chile, 2020.

- [19] M. Hernández-Rivera and C. Meza, “Quadratic adjustment with vertex convergence: An alternative algorithm for maximum power point tracking in photovoltaic modules,” in *2017 IEEE 37th Central America and Panama Convention (CONCAPAN XXXVII)*, 2017, pp. 1–6.
- [20] C. Meza, J. J. Negroni, F. Guinjoan, and D. Biel, “Inverter configurations comparative for residential pv-grid connected systems,” in *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*, 2006, pp. 4361–4366.
- [21] M. Hernández and C. Meza, “Descripción y simulación de sistemas fotovoltaicos utilizando modelica y open modelica,” Instituto Tecnológico de Costa Rica, Costa Rica, Tech. Rep., 2017.
- [22] A. Sánchez, “Desarrollo de una librería multicuerpo con ecosimpro,” B.S. Thesis, Universidad Carlos III de Madrid, España, 2011.
- [23] F. J. T. Valle, “Síntesis y descripción de circuitos digitales utilizando vhdl,” *Universidad Autónoma de Guadalajara*, 2001.
- [24] E. M. Clarke, M. Fujita, S. P. Rajan, T. Reps, S. Shankar, and T. Teitelbaum, “Program slicing of hardware description languages,” in *Advanced Research Working Conference on Correct Hardware Design and Verification Methods*. Springer, 1999, pp. 298–313.
- [25] M. Varo, “Comparativa didáctica de herramientas de simulación de circuitos analógicos,” B.S. Thesis, Universidad Politécnica de Madrid, España, 2017.
- [26] G. Rossum and the Python development team, “Python tutorial,” Python Software Foundation, Tutorial 3.8.3, May 2020.
- [27] C. Alexandru, J. Merchante, S. Panichella, S. Proksch, H. Gall, and G. Robles, “On the usage of pythonic idioms,” University of Zurich, Suiza, Rep., 2018.
- [28] M. Nosrati, “Python: An appropriate language for real world programming,” *World Applied Programming*, vol. 1, no. 2, pp. 110–117, June 2011.
- [29] W. F. Holmgren, C. W. Hansen, and M. A. Mikofski, “pvlib python: a python package for modeling solar energy systems,” *Journal of Open Source Software*, vol. 3, no. 29, p. 884, 2018. [Online]. Available: <https://doi.org/10.21105/joss.00884>
- [30] H. V., M. H., P. N., and D. W., *Ngspice User’s Manual*, Berkeley, 2021, release 34 plus. [Online]. Available: <http://ngspice.sourceforge.net/docs/ngspice-manual.pdf>

- [31] M. Fowler and K. Scott, *UML gota a gota: actualizado para cubrir la version 1*. Pearson Educación, 1999, vol. 2.
- [32] I. y. B. G. Rumbaugh, James y Jacobson, “El lenguaje de modelado unificado,” *Manual de referencia*.
- [33] X. F. Grau and M. I. S. Segura, “Desarrollo orientado a objetos con uml,” vol. 1, 2008.

## Abreviaturas

<b>ITCR</b>	Instituto Tecnológico de Costa Rica
<b>kW</b>	kilovatio
<b>IRENA</b>	Agencia Internacional de las Energías Renovables
<b>CWI</b>	Centrum Wiskunde & Informatica
<b>SESLab</b>	Laboratorio de Sistemas Eléctricos para la Sostenibilidad
<b>CC</b>	Corriente Continua
<b>CA</b>	Corriente Alterna
<b>SFCR</b>	Sistema Fotovoltaico Conectado a Red
<b>CD</b>	Corriente Directa
<b>MOSFET</b>	Metal-Oxide-Semiconductor Field-Effect Transistor
<b>CFFI</b>	C Foreign Function Interface
<b>API</b>	Application Programming Interface
<b>GUI</b>	Graphical User Interface
<b>LTS</b>	Long Term Support
<b>RAM</b>	Random Access Memory
<b>GB</b>	Gigabyte
<b>CPU</b>	Central Processing Unit
<b>UML</b>	Unified Modeling Language